



TUGAS AKHIR -SM141501

**Pengenalan Citra Digital Rambu Lalu Lintas
dengan Metode *Template Matching* Berbasis
*Color Moment***

YUSUF
NRP 1212 100 093

Dosen Pembimbing
Dr. Dwi Ratna Sulistyaningrum, S.Si, MT

JURUSAN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2016



FINAL PROJECT -SM141501

***TRAFFIC SIGN RECOGNITION USING TEMPLATE
MATCHING METHOD BASED ON COLOR MOMENT***

YUSUF
NRP 1212 100 093

Supervisors:
Dr. Dwi Ratna Sulistyaningrum, S.Si, MT

DEPARTMENT OF MATHEMATICS
Faculty of Mathematics and Natural Sciences
Sepuluh Nopember Institute of Technology
Surabaya 2016

LEMBAR PENGESAHAN

Pengenalan Citra Digital Rambu Lalu Lintas dengan Metode *Template Matching* Berbasis *Color Moment*


Traffic Sign Recognition Using Template Matching Method Based on Color Moment

TUGAS AKHIR


Diajukan untuk memenuhi salah satu syarat
Untuk memperoleh gelar Sarjana Sains
Pada bidang studi Ilmu Komputer
Program Studi S-1 Jurusan Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :
YUSUF
NRP. 1212100093

Menyetujui,
Dosen Pembimbing,


Dr. Dwi Ratna Sulistyaningrum, S.Si, MT
NIP. 19690405 199403 2 003

Mengetahui,
Ketua Jurusan Matematika
FMIPA-ITS


Dr. Imam Mukhlash, S.Si, MT
NIP. 19700831 199403 1 003
Surabaya, Juli 2016

Pengenalan Citra Digital Rambu Lalu Lintas dengan Metode *Template Matching* Berbasis *Color Moment*

Nama Mahasiswa : Yusuf
NRP : 1212 100 093
Jurusan : Matematika
Dosen Pembimbing : Dr. Dwi Ratna Sulistyaningrum, S.Si, MT

Abstrak

Sistem pengenalan rambu lalu lintas saat ini sedang menjadi topik menarik untuk dikembangkan. Sistem pengenalan rambu lalu lintas adalah bagian kecil dari sistem asisten kendali otomatis pada kendaraan roda empat. *Template Matching* adalah metode pengenalan rambu lalu lintas yang mampu mengatasi masalah sistem pengenalan yaitu dengan membandingkan dua citra menggunakan pengukuran jarak antara citra template acuan dan citra uji. Metode ini cocok untuk dikembangkan dalam sistem pengenalan rambu lalu lintas secara real time. Keunggulan dari metode ini pada tugas akhir ini adalah membandingkan fitur vektor 2 citra yang telah melalui proses ekstraksi fitur vektor yang secara efisien dapat mengenali citra rambu lalu lintas. Pada tugas akhir ini telah dilakukan pengenalan citra digital rambu lalu lintas dengan metode *Template Matching*. Sistem ini dibagi menjadi 2 tahapan, yaitu tahap deteksi dan tahap pengenalan. Tahap deteksi objek rambu lalu lintas pada penelitian ini menggunakan metode segmentasi warna *hue, saturation, value* dan ROI (*Region of Interest*). Tahap selanjutnya, citra kandidat diteruskan ke tahap pengenalan dengan menggunakan metode *template matching*. Metode ini mempunyai konsep mencocokkan antara 2 citra yang mempunyai ukuran sama untuk dihitung jarak kemiripan (*similarity measurement*). Pada tahap pengenalan, baik antara citra uji dan citra template acuan dilakukan proses ekstraksi fitur vektor menggunakan metode *color moment*. Ekstrak fitur vektor dari citra tersebut yang dihitung jarak

kemiripannya menggunakan metode jarak *euclidean*. Hasil pengujian menunjukkan bahwa sistem dapat mengenali citra rambu lalu lintas dengan akurasi 96.69%.

Kata Kunci—Pengenalan Rambu Lalu Lintas, segmentasi HSV, ROI, *color moment*, *template matching*.

TRAFFIC SIGN RECOGNITION USING TEMPLATE MATCHING METHOD BASED ON COLOR MOMENT

Name : Yusuf
NRP : 1212 100 093
Department : Mathematics
Supervisor : Dr. Dwi Ratna S, S.Si, MT

Abstract

Traffic sign recognition system is being an interested topic to be developed in nowadays. It is a small part of Autonomous Driver Assistance Systems (ADAS). Template matching is a method for recognizing system which could cover its matter by comparing the distance calculation of two images. This method is good way for developing the real-time traffic sign recognition system. The benefits using this method is just comparing the vector features of those 2 images which had going through from vector features extraction process, so it can recognizing the traffic sign images efficiently. Template matching method had been used to recognized some traffic sign image in it. The system is divide into two phases, detetction and recognition. We proposed hue, saturation, values method and Region of Interest (ROI) for detecting traffic sign object in detection phase. On the next phase, candidate image forwarded to recognition phase by using template matching method. This method has the concept of a match between two images that have same size to be calculated similarity distances. In recognition phase, both test images and template images has gone through the process of vector feature extraction using color moment method. These vector feature extract that calculated the similarity distance using euclidean distance. The test result show that system can recognize traffic sign images with an accuracy of 96.69%.

Keywords: *Traffic Sign Recognition, HSV Segmentation, ROI, color moment, template matching.*

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
LEMBAR PENGESAHAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR LAMPIRAN	xxi
BAB I. PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan	4
1.5 Manfaat.....	4
1.6 Sistematika Penulisan Tugas Akhir	4
BAB II. KAJIAN TEORI	
2.1 Penelitian Sebelumnya.....	7
2.2 Pengertian Citra	8
2.3 <i>Color Moment</i>	10
2.4 <i>Region of Interest (ROI)</i>	11
2.5 <i>Template Matching</i>	12
2.6 Rambu Lalu Lintas	13
BAB III. METODOLOGI	
3.1 Objek Penelitian.....	15
3.2 Peralatan	15
3.3 Tahap Penelitian	15
BAB IV. PERANCANGAN DAN IMPLEMENTASI SISTEM	
4.1 Analisis Sistem	19
4.1.1 Analisis Sistem Perangkat Lunak	19
4.1.2 Analisis Kebutuhan Sistem.....	26
4.2 Perancangan Sistem	27

4.2.1	Gambaran Sistem.....	27
4.2.2	Penjelasan Umum Sistem	30
4.2.2.1	Perancangan Data	30
4.2.2.2	Data Masukan	31
4.2.2.3	Data Proses	31
4.2.2.3	Data Luaran	31
4.3	Penjelasan Proses Sistem	31
4.3.1	Proses Deteksi.....	32
4.3.2	<i>Region of Interest</i>	34
4.3.3	Proses Perubahan Skala Resolusi	35
4.3.4	Pembangunan Dataset.....	36
4.3.5	Ekstraksi Fitur	37
4.3.6	<i>Template Matching</i>	38
4.4	Implementasi Sistem.....	38
4.4.1	Implementasi Antarmuka.....	39
4.4.1.1	Halaman Utama	39
4.4.1.2	Antarmuka Deteksi	40
4.4.1.3	Antarmuka Pengenalan.....	41
4.4.2	Implementasi Tahap Input Data.....	43
4.4.3	Implementasi Proses Segmentasi HSV.....	43
4.4.4	Implementasi Proses ROI	44
4.4.5	Implementasi Pembangunan Dataset	45
4.4.6	Implementasi Pemilihan Citra Uji	46
4.4.6	Implementasi Ekstraksi Fitur Citra Uji	46
4.4.6	Implementasi Pengukuran Jarak	47
4.4.6	Implementasi Proses Pengenalan	48
4.4.6	Implementasi Penampilan Informasi	49

BAB V. PENGUJIAN DAN PEMBAHASAN

5.1	Data Uji Coba	51
5.2	Pengujian Tahap Deteksi	52
5.3	Pembahasan Konversi RGB ke HSV	57
5.4	Pengujian Tahap ROI	58
5.5	Pembahasan Pengubahan Skala Resolusi	58
5.6	Pengujian Ekstraksi Fitur	59
5.7	Pengujian Tahap Pembangunan Dataset	60

5.8 Pengujian Tahap Pengenalan	60
5.9 Pembahasan Perhitungan Jarak	68
5.10 Pembahasan Hasil Pengujian	71
5.11 Pembahasan Penyebab Besar Kecilnya Akurasi ..	71
BAB VI. KESIMPULAN DAN SARAN	
6.1 Kesimpulan	73
6.2 Saran	73
DAFTAR PUSTAKA	75
LAMPIRAN	79

DAFTAR GAMBAR

	Halaman
Gambar 2.1	Contoh beberapa gambar rambu lalu lintas..... 13
Gambar 3.1	Diagram Proses Deteksi 17
Gambar 3.2	Diagram Proses Tahap Pengenalan 18
Gambar 4.1	<i>Use Case Diagram</i> Pengenalan Rambu Lalu Lintas 20
Gambar 4.2	<i>Swimlane Diagram</i> Pengenalan 21
Gambar 4.3	Citra input dan citra hasil segmentasi HSV..... 23
Gambar 4.4	Citra ROI dan hasilnya..... 23
Gambar 4.5	Kumpulan data citra terdeteksi 24
Gambar 4.6	Proses penyimpatan dataset fitur vektor..... 25
Gambar 4.7	Proses <i>template matching</i> 26
Gambar 4.8	Diagram alir sistem deteksi..... 28
Gambar 4.9	Diagram alir sistem pengenalan 29
Gambar 4.10	Diagram alir proses deteksi 33
Gambar 4.11	Diagram alir proses ROI 34
Gambar 4.12	Diagram alir proses <i>Rescalling</i> 35
Gambar 4.13	Diagram alir proses pembuatan dataset..... 36
Gambar 4.14	Diagram alir proses ekstraksi fitur 37
Gambar 4.15	Diagram alir proses <i>template matching</i> 38
Gambar 4.16	Antarmuka halaman utama..... 39
Gambar 4.17	Antarmuka deteksi 41
Gambar 4.18	Antarmuka pengenalan..... 42
Gambar 5.1	Contoh beberapa data citra masukan..... 52
Gambar 5.2	Contoh data citra input 53
Gambar 5.3	Citra <i>Hue, Saturation, Value</i> 53
Gambar 5.4	Citra hasil HSV <i>masked</i> dan citra gabungannya 54
Gambar 5.5	Citra reduksi <i>smallest area</i> 54
Gambar 5.6	Citra RGB <i>masking</i> 55
Gambar 5.7	Contoh citra input gagal terdeteksi..... 55
Gambar 5.8	Citra HSV 56
Gambar 5.9	Citra HSV <i>masked</i> 56
Gambar 5.10	Citra hasil gagal terdeteksi 57

Gambar 5.11	Citra hasil ROI.....	58
Gambar 5.12	Citra hasil <i>rescalling</i>	59
Gambar 5.13	Citra input.....	68
Gambar 5.14	Citra kandidat deteksi dan <i>template</i>	69
Gambar 5.15	Hasil jarak <i>euclidean</i>	69
Gambar 5.15	Hasil jarak <i>normalized euclidean</i>	69

DAFTAR TABEL

	Halaman
Tabel 4.1 Tabel Kebutuhan Sistem.....	26
Tabel 4.2 Tabel Kegunaan Menu Sistem.....	40
Tabel 5.1 Hasil Ekstrak Fitur <i>Color Moment</i>	59
Tabel 5.2 Hasil Pembangunan Dataset	60
Tabel 5.3 Hasil Pengujian Deteksi dan Pengenalan.....	61
Tabel 5.4 Pengelompokan Data Citra berdasarkan Info	66
Tabel 5.5 Pengaruh Jumlah Data Acuan terhadap Akurasi	67

DAFTAR LAMPIRAN

	Halaman
Lampiran A : Data Citra	79
Lampiran B : Citra Kandidat Terdeteksi	111
Lampiran C : Kode Program Deteksi	115
Lampiran D : Kode Program Pengenalan	123

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pada zaman era modern saat ini teknologi telah berkembang sangat pesat. Di bidang otomotif khususnya mobil banyak implementasi teknologi yang sudah diterapkan kedalam kendaraan tersebut. Salah satu teknologi yang saat ini diterapkan dan sedang berkembang dalam dunia otomotif adalah ADAS (*Advanced Driver Assistance Systems*) [19]. ADAS adalah sebuah sistem yang dapat mengenali rambu lalu lintas di jalan raya dan kemudian memberitahu pengemudi mengenai arti dari rambu-rambu lalu lintas tersebut untuk mengendalikan batas kecepatan, adanya peringatan, adanya bahaya, dan lainnya. Teknologi ini mempunyai kemampuan untuk meningkatkan keamanan bagi pengendara dengan selalu mengawasi pengendara dan lingkungan sekitarnya. Kendala utama dari sistem ini adalah sistem harus dapat mengenali kondisi kendaraan saat melaju cepat. Selain itu sistem ini harus tanggap dengan segala perubahan cuaca lingkungan yang ada dan tetap bisa bekerja akurat secara terus menerus (*real time*). ADAS juga memiliki tujuan dasar untuk membantu dan mengarahkan pengendara pada saat mengendarai kendaraan sehingga dapat meningkatkan keselamatan lalu lintas, efisiensi lalu lintas, dan perbaikan kondisi lingkungan. Berbagai metode terkait pengenalan citra rambu lalu lintas yang mendukung sistem tersebut telah banyak diterapkan di ADAS. Beberapa diantaranya yang telah dikembangkan adalah *SVM Method* [17], *HOG features* [17], *Ramer-Douglas-Peucker algorithm* [17], *SURF algorithm* [16], *Template Matching* [4].

Template Matching adalah metode pengenalan yang digunakan untuk mengenali objek rambu lalu lintas di sebuah citra berdasarkan *template* yang telah ditentukan lalu melakukan identifikasi sebagai bentuk pengenalan citra [4].

Pada Tahun 2011, David Soendoro dan Iping Supriana melakukan penelitian tentang *Traffic Sign Recognition* menggunakan metode *Color-based Method* dan *Shape-arc* untuk proses deteksi bentuk dan warna dari rambu lalu lintas dan *SVM Method* untuk klasifikasi rambu lalu lintas [17]. Pada penelitiannya terdapat kendala masalah pencahayaan. Penelitian yang berjudul “*Traffic Sign Recognition*” yang dilakukan oleh Chetan dan Pravin pada tahun 2012 mengenai pengenalan rambu lalu lintas dengan 3 tahapan yaitu segmentasi, deteksi, dan pengenalan [16]. Metode yang digunakan adalah *Color Segmentation* untuk segmentasi, *Region Of Interest (ROI)* untuk deteksi, dan *SVM method* untuk pengenalan. Penelitian tersebut masih terkendala oleh beberapa permasalahan, seperti sistem yang telah dibangun masih sulit untuk mendeteksi rambu lalu lintas berdasarkan segmentasi warna jika pada suatu gambar ada kesamaan warna dengan objek lain yang sama dengan warna rambu lalu lintas. Penelitian yang dilakukan oleh Sheldon dan Erdal yang berjudul “*FPGA-Based Traffic Sign Recognition for Advance Driver Assistance System*” membahas sebuah hardware dasar untuk ADAS[9]. Dalam penelitiannya Sheldon dan Erdal terdiri dari 2 proses pengolahan citra, yaitu deteksi dan pengenalan. Diakhir penelitian tersebut terdapat permasalahan dalam deteksi menggunakan *Hue Calculation* yang dirasa peneliti kurang efisien untuk proses deteksi. Penelitian pada tahun 2014 yang berjudul “*Traffic Sign Recognition – How far are we from the solution?*” yang dilakukan oleh Markus, Radu, Rodrigo, dan Luc ini membandingkan beberapa metode pengenalan rambu lalu lintas yang ada di Jerman dan Belgia untuk ditarik kesimpulan metode mana yang paling efektif, hasil dari metode yang ada, dijumpai berbagai masalah terutama dalam proses *computational time* yang dirasa peneliti lama dan kurang efisien dalam *computational cost* [9]. Tugas Akhir yang dilakukan oleh Mira Aryani dan Nurul Hidayat dengan judul “Pengenalan Gambar Rambu Lalu Lintas menggunakan

Metode *Template Matching*” [2]. Pada penelitiannya masih ditemukan kekurangan atau kelemahan pada penggunaan metode *color thresholding* dan metode proyeksi vertikal dan horizontal yaitu tidak dapat mendeteksi dua objek dengan warna yang sama. Pada metode kuantisasi vector juga memiliki kelemahan yaitu jika segmen yang diambil kecil maka waktu komputasinya semakin besar.

Berdasarkan masalah diatas, penulis tertarik untuk melakukan penelitian pengenalan citra digital rambu lali lintas yang ada di Indonesia dengan menerapkan metode *Template Matching*.

1.2 Rumusan Masalah

Rumusan masalah dari Tugas Akhir ini adalah :

1. Bagaimana melakukan deteksi objek yang berisi rambu lalu lintas pada citra digital dengan metode segmentasi warna *hue, saturation, value*.
2. Bagaimana melakukan pengenalan sebuah objek rambu lalu lintas dengan metode *template matching*.

1.3 Batasan Masalah

Batasan Masalah yang akan dibahas dalam penelitian Tugas Akhir ini adalah sebagai berikut :

1. File citra RGB yang digunakan dalam bentuk format .JPG.
2. Bentuk rambu lalu lintas digunakan untuk pengenalan adalah rambu lalu lintas yang berbentuk belah ketupat dan lingkaran karena frekuensi kemunculan di jalan sangat sering.
3. *Software* yang digunakan dalam implementasi penelitian ini adalah MATLAB.
4. Rambu lalu lintas yang digunakan acuan adalah rambu lalu lintas yang ada di Indonesia.
5. Citra masukkan adalah citra yang memuat *single* objek rambu lalu lintas.

1.4 Tujuan

Tujuan dari Tugas Akhir ini adalah :

1. Melakukan deteksi objek yang berisi rambu lalu lintas pada citra digital dengan metode segmentasi warna *hue, saturation, value*.
2. Melakukan pengenalan sebuah objek rambu lalu lintas dengan metode *template matching*

1.5 Manfaat

Manfaat Tugas Akhir ini untuk membantu pihak-pihak yang terkait pengembang pengenalan rambu lalu lintas untuk sistem kendali pengendara otomatis kendaraan bermotor, sebagai pembanding dengan penelitian terkait tentang penggunaan metode yang berbeda pada sistem kendali pengendara otomatis. Selain itu, sebagai bahan refrensi untuk pengembangan peneletian selanjutnya di kemudian hari.

1.6 Sistematika Penulisan Tugas Akhir

Sistematika penulisan didalam Tugas Akhir ini adalah sebagai berikut:

1. BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang pembuatan Tugas Akhir, rumusan dan batasan permasalahan yang dihadapi dalam penelitian Tugas Akhir, tujuan dan manfaat pembuatan tugas akhir dan sistematika penulisan tugas akhir.

2. BAB II KAJIAN TEORI

Bab ini menjelaskan tentang kajian teori dari referensi penunjang serta penjelasan permasalahan yang dibahas dalam Tugas Akhir ini, meliputi Pengertian Citra Digital, Citra RGB dan *grayscale*, *Region Of Interest*, Rambu lalu lintas, mode warna HSV, dan Algoritma *template matching* untuk proses pengenalan citra.

3. **BAB III METODOLOGI PENELITIAN**

Bab ini berisi metodologi atau urutan pengerjaan yang dilakukan dalam menyelesaikan Tugas Akhir, meliputi studi literatur, pengumpulan data, analisis dan desain sistem, pembuatan program, uji coba dan evaluasi, hingga penulisan Tugas Akhir.

4. **BAB IV PERANCANGAN DAN IMPLEMENTASI**

Bab ini menjelaskan tentang perancangan dan implementasi sistem pengenalan citra digital rambu lalu lintas dengan metode *template matching*, serta proses pembuatan sistem secara keseluruhan.

5. **BAB V PENGUJIAN DAN PEMBAHASAN HASIL**

Bab ini akan menampilkan hasil uji coba serta pengujian terhadap parameter yang telah digunakan dan selanjutnya digunakan dalam perumusan kesimpulan dan saran.

6. **BAB VI PENUTUP**

Bab ini merupakan penutup, berisi tentang kesimpulan yang dapat diambil berdasarkan data yang ada dan saran yang selayaknya dilakukan bila Tugas Akhir ini dilanjutkan.

BAB II

KAJIAN TEORI

2.1 Penelitian Sebelumnya

Penelitian terkait Tugas Akhir ini tentang pengenalan rambu lalu lintas telah dilakukan oleh Markus Mathias dkk yang berjudul *Traffic Sign Recognition – How far are we from the solution?* [9]. Dalam penelitian tersebut mereka menggunakan metode variansi modern dari *HOG features* untuk deteksi dan *sparse representations* untuk pengenalan rambu lalu lintas yang ada di Jerman dan Belgia. Berdasarkan hasil penelitian tersebut metode pengenalan rambu lalu lintasnya mencapai tingkat akurasi sampai 95%.

David Soendoro dan Iping Supriana telah melakukan penelitian terkait pengenalan rambu lalu lintas yang berjudul *Traffic Sign Recognition with Color-based Method, Shape-arc Estimation and SVM* [17]. Dalam jurnal yang dituliskannya dijelaskan bahwa sistem yang dibangun untuk pengenalan rambu lalu lintas menggunakan *Color-based Method* dan *Ramer-Douglas-Peucker Algorithm* untuk deteksi objek rambu lalu lintas dan menggunakan *SVM Method* untuk pengenalan citra biner rambu lalu lintas. Hasil dari penelitian tersebut mendapatkan nilai akurasi sampai 97%.

Penelitian tentang metode *template matching* untuk pengenalan rambu lalu lintas telah digunakan oleh Sheldon Waite dan Erdal Oruklu pada penelitian yang berjudul *FPGA- Based Traffic Sign Recognition for Advance Driver Assistance Systems* [19]. Dalam penelitiannya, Waite membagi proses pengolahan citra menjadi dua proses, meliputi deteksi dan pengenalan. Dalam proses deteksi metode yang digunakan adalah *Hue Calculation and Detection*, dan *Morphological Filter*. Pada proses pengenalan menggunakan metode *labelling*, *candizate resizing* dan *template matching*.

Tugas Akhir tentang pengenalan gambar rambu lalu lintas menggunakan metode *template matching* telah dilakukan oleh Mira Aryani dan Nurul Hidayat [2]. Dalam buku Tugas Akhir yang dituliskannya, penulis menjelaskan bahwa sistem yang dibangun untuk proses deteksi menggunakan metode *color thresholding* dan metode proyeksi vertikal dan horizontal yang mana dalam implementasinya masih banyak kekurangan yaitu tidak dapat mendeteksi rambu dengan warna yang sama jika ada objek lain disekitarnya lalu sistem juga masih belum bisa mengatasi pengenalan secara real time. Metode kuantisasi vektor digunakan untuk pengambilan segmen deteksi dalam *channel* warna *grayscale*. Pada proses pengenalan dengan metode *template matching* menghasilkan prosentasi keberhasilan sebesar 78%. Dalam hal ini perbedaan dengan Tugas Akhir yang dikerjakan pada kali ini adalah proses deteksi menggunakan segmentasi HSV dan citra tidak perlu dikonversi ke *grayscale* yang nantinya akan merubah *channel* warnanya. Metode *template matching* yang digunakan pada Tugas Akhir ini menggunakan perbandingan fitur vektor *color moment* dari citra sedangkan yang digunakan oleh Mira Ariyani menggunakan *blocking* yang dirasa penulis kurang efisien dalam penggunaannya.

2.2. Pengertian Citra

Citra menurut Kamus Besar Bahasa Indonesia memiliki makna rupa, gambar, atau gambaran. Citra terbagi menjadi dua yaitu citra diam dan citra bergerak. Citra diam adalah citra tunggal yang tidak bergerak. Sedangkan, citra bergerak adalah rangkaian citra diam yang ditampilkan secara beruntun sehingga memberi kesan pada mata kita sebagai gambar yang bergerak.

Citra RGB (*Red- Green-Blue*) merupakan warna dasar dari suatu citra warna. Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari ketiga warna dasar RGB. Setiap titik pada citra warna membutuhkan data sebesar 3 byte. Setiap warna dasar memiliki intensitas tersendiri dengan nilai minimum nol (0) dan nilai maksimum 255 (8 bit). RGB didasarkan pada

teori bahwa mata manusia peka terhadap panjang gelombang 630 nm (merah), 530 nm (hijau), dan 450 nm (biru).

Hue, Saturation, Value (HSV) adalah satu dari beberapa sistem warna atau *channel* warna yang digunakan orang untuk memilih warna (misalnya cat atau tinta) dari *color wheel*. Model warna ini lebih dipandang baik daripada model RGB sebagai cara dimana manusia mencoba dan menggambarkan sensasi warna. Metode ini berguna untuk mendeteksi citra rambu lalu lintas berdasarkan segmentasi *channel* warnanya, sehingga objek rambu lalu lintas dalam sebuah citra dapat tersegmentasi warnanya dari objek lain yang berbeda warna dalam suatu *frame* citra. Persamaan (2.1), (2.2), dan (2.3) menunjukkan cara konversi nilai RGB ke HSV yang digunakan untuk proses deteksi [13] :

$$H = \begin{cases} \text{Tidak terdefinisi, jika } \max = \min \\ \frac{60^\circ \times \frac{G-B}{\max-\min} + 0^\circ}{360}, \text{ jika } \max = R \text{ dan } G \geq B \\ \frac{60^\circ \times \frac{G-B}{\max-\min} + 360^\circ}{360}, \text{ jika } \max = R \text{ dan } G < B \\ \frac{60^\circ \times \frac{G-B}{\max-\min} + 120^\circ}{360}, \text{ jika } \max = G \\ \frac{60^\circ \times \frac{G-B}{\max-\min} + 240^\circ}{360}, \text{ jika } \max = B \end{cases} \quad (2.1)$$

$$S = \begin{cases} 0, & \text{jika } \max = 0 \\ 1 - \left(\frac{\min}{\max}\right), & \text{untuk lainnya} \end{cases} \quad (2.2)$$

$$V = \frac{\max}{255} \quad (2.3)$$

H = Nilai hue

S = Nilai saturation

V = Nilai value

2.3. *Color Moment*

Color moment merupakan representasi padat dari fitur warna dalam mengkarakterisasikan warna gambar. Perhitungan momen digunakan untuk mendapatkan *color similarity* sebuah citra dimana nilai dari *similarity* tersebut digunakan untuk membandingkan citra uji dengan citra yang terdapat pada dataset citra. Metode ini hanya untuk mengekstraksi dan menyimpan fitur vektor dominan pada distribusi warna di dalam dataset citra. Ruang warna yang digunakan adalah RGB yang terdiri dari tiga komponen warna (*Red, Green, Blue*), dimana fitur vektor citra dibagi menjadi 9 momen, yaitu 3 momen untuk masing-masing 3 *color channels* (*Red, Green, Blue*) sehingga akan menghasilkan 9 ekstraksi fitur vektor. Kemudian 9 fitur vektor ini dianggap sebagai ciri warna dari sebuah citra. 3 momen tersebut adalah *mean, standard deviation*, dan *skewness*. Metode ini digunakan dalam proses pengenalan di tahap ekstraksi fitur vector, baik untuk citra template acuan dan citra uji. Ketiga momen warna tersebut didefinisikan oleh persamaan (2.4), (2.5), (2.6) berikut [13] :

$$E_k = \frac{1}{N} \sum_{i=1}^m \sum_{j=1}^n p_{ijk}, \quad k = 1, 2, 3 \quad (2.4)$$

Mean bisa diartikan sebagai rata-rata nilai warna dari suatu citra.

$$\sigma_k = \sqrt{\left(\frac{1}{N} \sum_{i=1}^m \sum_{j=1}^n (p_{ijk} - E)^2 \right)}, \quad k = 1, 2, 3 \quad (2.5)$$

Standard Deviasi adalah akar dari nilai variansi distribusi warnanya.

$$s_k = \sqrt[3]{\left(\frac{1}{N} \sum_{i=1}^m \sum_{j=1}^n (p_{ijk} - E)^3\right)}, \quad k = 1, 2, 3 \quad (2.6)$$

Skewness dapat diartikan sebagai sebuah ukuran dari derajat asimetri dari distribusinya.

N = Ukuran citra m x n

E = Mean

p_{ij} = nilai channel warna – k di piksel citra (i, j)

σ = Standar Deviasi

s = Skewness

k = *channel* warna *red, green, blue*

n = Banyaknya komponen j

m = Banyaknya komponen i

2.4. *Region of Interest (ROI)*

Region Of Interest (ROI) adalah bagian wilayah dari suatu gambar yang ingin dipilih untuk dilakukan beberapa operasi. ROI bisa didefinisikan dengan menciptakan tanda biner, citra biner yang merupakan ukuran yang sama dengan gambar yang dikelola dengan piksel yang mendefinisikan pengaturan/set ROI ke angka 1 dan semua piksel lainnya diatur ke angka 0. Misalnya pemilihan daerah geografis, pengambilan poligon yang mencakup piksel berdekatan, atau didefinisikan oleh berbagai intensitas. Pada tahap deteksi, ROI digunakan untuk memilih daerah tertentu dari suatu citra sehingga didapatkan objek citra rambu lalu lintas saja dalam satu *frame* citra. Hasil citra yang telah melewati proses ROI akan mudah dikenali pada saat tahap pengenalan.

2.5. *Template Matching*

Template Matching adalah sebuah algoritma yang membandingkan citra uji dengan citra *template* dataset yang sudah dikenali baik sebagai rambu lalu lintas sebagai acuannya. Setiap citra uji akan dibandingkan dengan setiap citra *template* dan jika saat ditemukan citra tersebut mendekati kecocokan maka akan dideklarasikan sebagai rambu lalu lintas. Persamaan (2.7) berikut adalah persamaan *template matching* dengan jarak *Euclidean* [19]:

$$d(\bar{x}, \bar{y}) = \sqrt{\sum_{i=1}^n (\bar{x}_i - \bar{y}_i)^2} \quad (2.7)$$

\bar{x} = Fitur vector ciri citra *template* acuan

\bar{y} = Fitur vektor ciri citra uji

$d(\bar{x}, \bar{y})$ = jarak *Euclidean* antara x dan y

\bar{x}_i = Nilai fitur vector ciri ke-i dari citra x

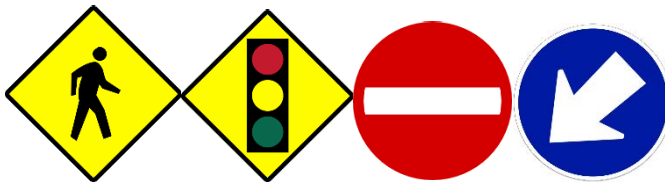
\bar{y}_i = Nilai fitur vektor ciri ke-i dari citra y

n = Banyaknya komponen i

Sebuah nilai kecil dari $d(x, y)$ mengindikasikan kesamaan pola. Dalam penelitian ini dilakukan penghitungan selisih jarak fitur vektor *color moment* antara citra *template* acuan dengan citra uji. Pada tahap pengenalan, fitur vektor *color moment* citra *template* acuan tersebut dicocokkan dengan fitur vektor *color moment* citra uji rambu lalu lintas. Selisih terkecil antara kedua fitur vektor *color moment* itu dipakai sebagai hasil pengenalan. Pencocokan tersebut menghasilkan prosentase akurasi kecocokan antara citra *template* dengan citra uji rambu lalu lintas.

2.6. Rambu Lalu Lintas

Rambu lalu lintas adalah suatu bagian dari perlengkapan jalan yang memuat lambang, huruf, angka, kalimat dan / atau perpaduan di antaranya, yang digunakan untuk memberikan peringatan, larangan, perintah dan petunjuk bagi pemakai jalan. Rambu lalu lintas diatur menurut Peraturan Menteri Perhubungan Nomor 13 tahun 2014. Tanda-tanda ini menyampaikan pesan dalam kata-kata atau symbol dan digunakan untuk mengatur, memperingatkan, atau memandu pengguna jalan (pengendara, pejalan kaki, dll). Rambu lalu lintas umum digunakan sebagai alat keselamatan lalu lintas, terutama dikembangkan untuk memberikan informasi penting dalam waktu singkat untuk mendukung pengguna jalan dengan aman



Gambar 2.1 Contoh beberapa gambar rambu lalu lintas

BAB III METODOLOGI

Pada bab ini dijelaskan metode penelitian Tugas Akhir yang digunakan untuk menyelesaikan permasalahan pada Tugas Akhir ini. Objek penelitian yang akan digunakan pada Tugas Akhir adalah citra digital rambu lalu lintas yang mempunyai format *.jpg. Peralatan penelitian yang digunakan untuk menyelesaikan tugas akhir ini adalah MATLAB sebagai perangkat lunak utama untuk membuat antarmuka dan membangun sistem pengenalan rambu lalu lintas.

3.1 Objek Penelitian

Objek penelitian yang akan digunakan pada Tugas Akhir adalah citra digital yang memuat objek rambu lalu lintas. Citra didapatkan langsung dari hasil pemotretan di jalan raya. Objek yang digunakan pada tugas akhir kali ini adalah citra yang single objek rambu lalu lintas. Bentuk rambu lalu lintas yang diteliti pada tugas akhir kali ini adalah rambu lalu lintas berbentuk belah ketupat dan lingkaran.

3.2 Peralatan

Peralatan penelitian yang digunakan untuk menyelesaikan tugas akhir ini adalah MATLAB sebagai perangkat lunak utama untuk membuat antarmuka dan membangun sistem pengenalan rambu lalu lintas dengan metode *template matching*. Proses pengambilan citra digital rambu lalu lintas menggunakan kamera digital beresolusi 13 megapixel dan dengan pengaturan mode pemotretan auto.

3.3 Tahap Penelitian

Adapun tahap-tahap yang dilakukan dalam penyusunan Tugas Akhir ini adalah sebagai berikut :

1. Studi Literatur

Pada tahap ini akan dikaji tentang pengolahan citra digital rambu lalu lintas dan konsep sistem pengenalan rambu lalu lintas. Studi ini dilakukan dengan membaca jurnal pengenalan rambu lalu lintas, jurnal deteksi dan yang berkaitan dengan penggunaan metode *Template Matching*.

2. Analisis Pengolahan Citra dan Implementasi program pengenalan citra rambu lalu lintas dengan *template matching*

Pada tahapan ini akan dilakukan analisis citra, perancangan gambaran umum sistem, perancangan proses algoritma, perancangan data, dan perancangan antar muka sistem. Citra akan dianalisis menggunakan metode Segmentasi *Hue, Saturation, Value* untuk mengekstraksi warna dan bentuk objek yang diinginkan. Selanjutnya citra yang telah terdeteksi menjadi citra kandidat rambu lalu lintas akan diteruskan ke proses pengenalan menggunakan metode *Template Matching* yang pada akhirnya akan dikenali citra tersebut sebagai rambu lalu lintas.

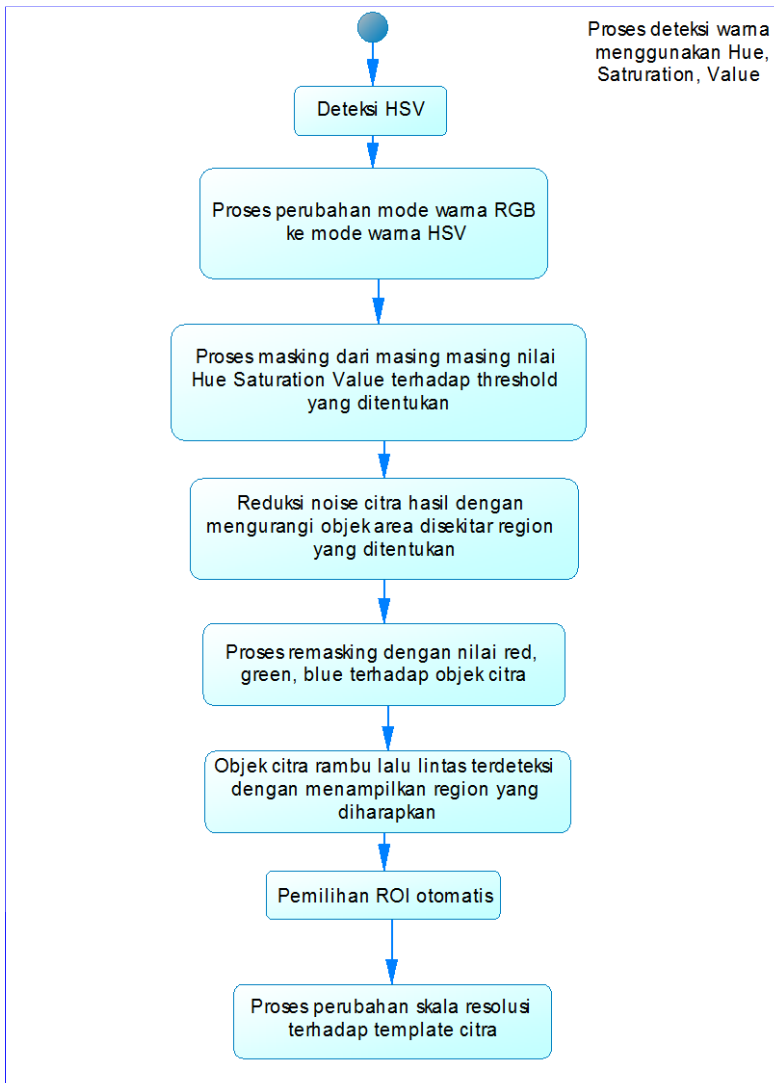
3. Uji Coba dan Analisis hasil eksperimen

Pada tahap ini akan dilakukan pengujian terhadap program yang telah dibangun. Pengujian ini menggunakan metode *template matching* yang dihitung berdasarkan nilai jarak *euclidean* antara citra uji dengan setiap citra *template* acuan yang ada.

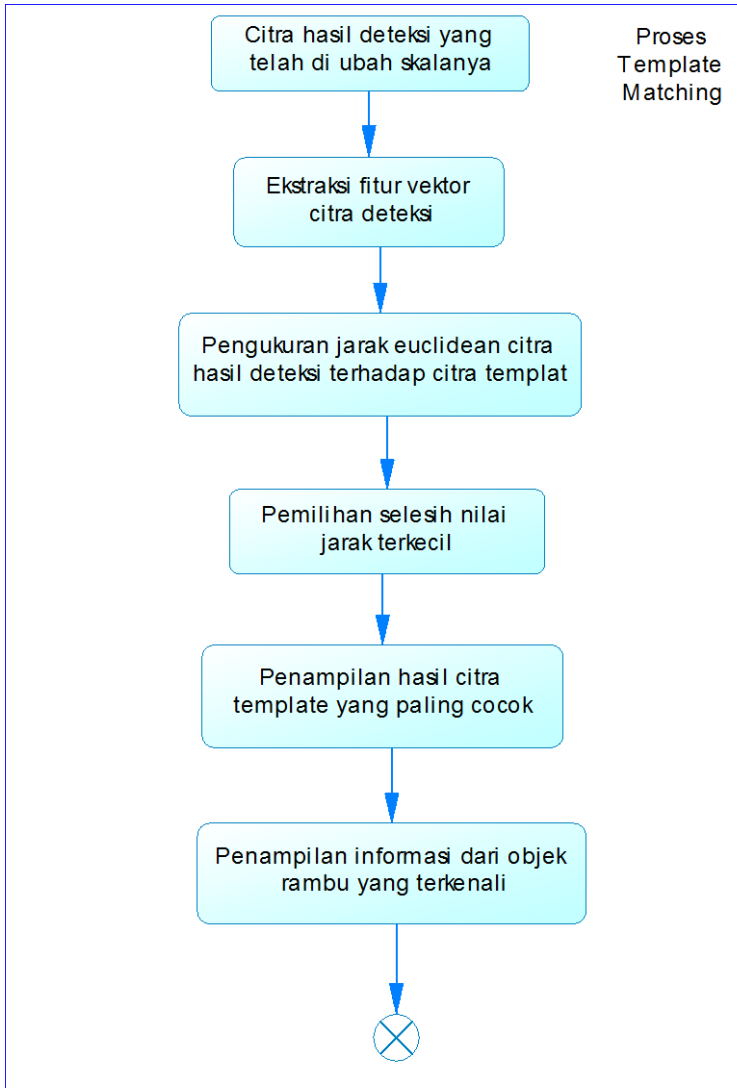
4. Penarikan Kesimpulan

Tahap penarikan kesimpulan merupakan tahap akhir dalam proses penelitian Tugas Akhir ini, dimana pada tahap ini dilakukan penarikan kesimpulan terhadap hasil yang telah dicapai.

Tahap-tahap pengerjaan Tugas Akhir yang telah dijelaskan di atas digambarkan dalam diagram alir pada Gambar 3.1 dan 3.2



Gambar 3.1. Diagram Proses Deteksi



Gambar 3.2. Diagram Proses Tahap Pengujian Pengenalan dengan *template matching*

BAB IV

PERANCANGAN DAN IMPLEMENTASI SISTEM

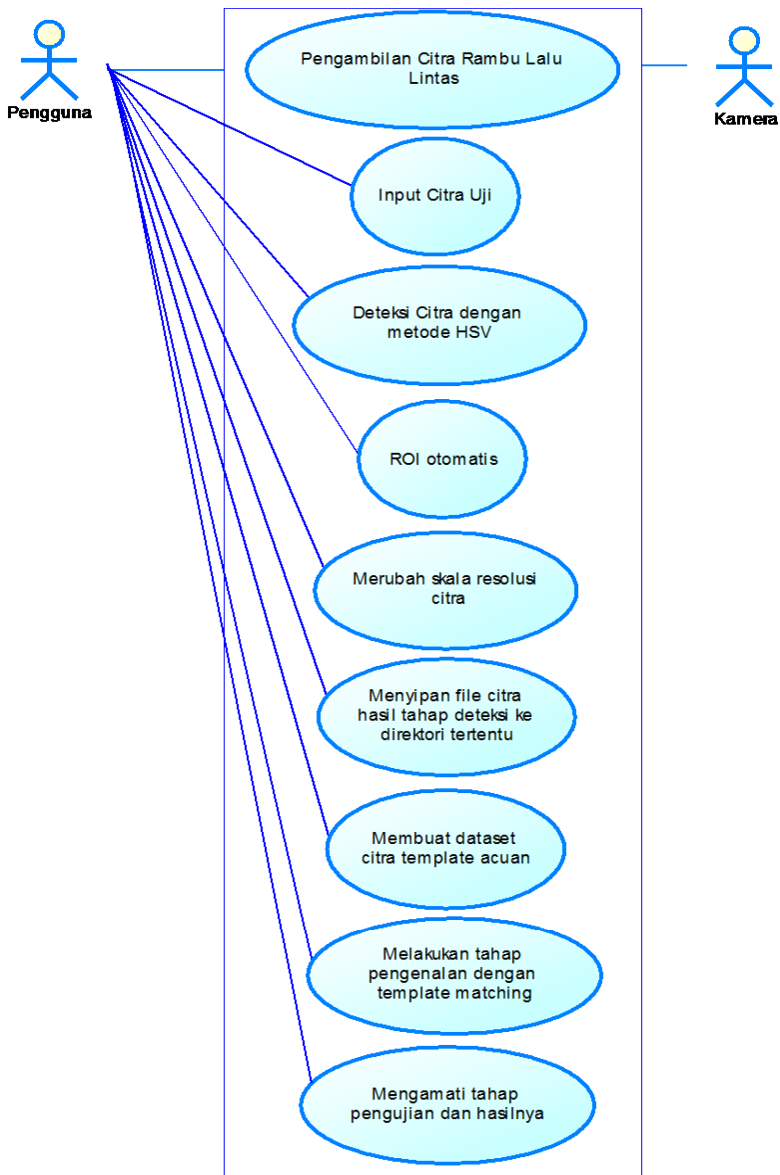
Pada bab ini akan dibahas mengenai perancangan dan implementasi sistem dimulai dari pembahasan proses pengambilan citra, perancangan dataset objek, deteksi objek rambu lalu lintas menggunakan metode segmentasi HSV, pengenalan citra menggunakan metode *Template Matching*, perancangan GUI(*Graphic User Interface*) pada MATLAB, serta penjelasan mengenai cara untuk mendapatkan data keluaran yang sesuai dengan tujuan dari penelitian Tugas Akhir ini.

4.1 Analisis Sistem

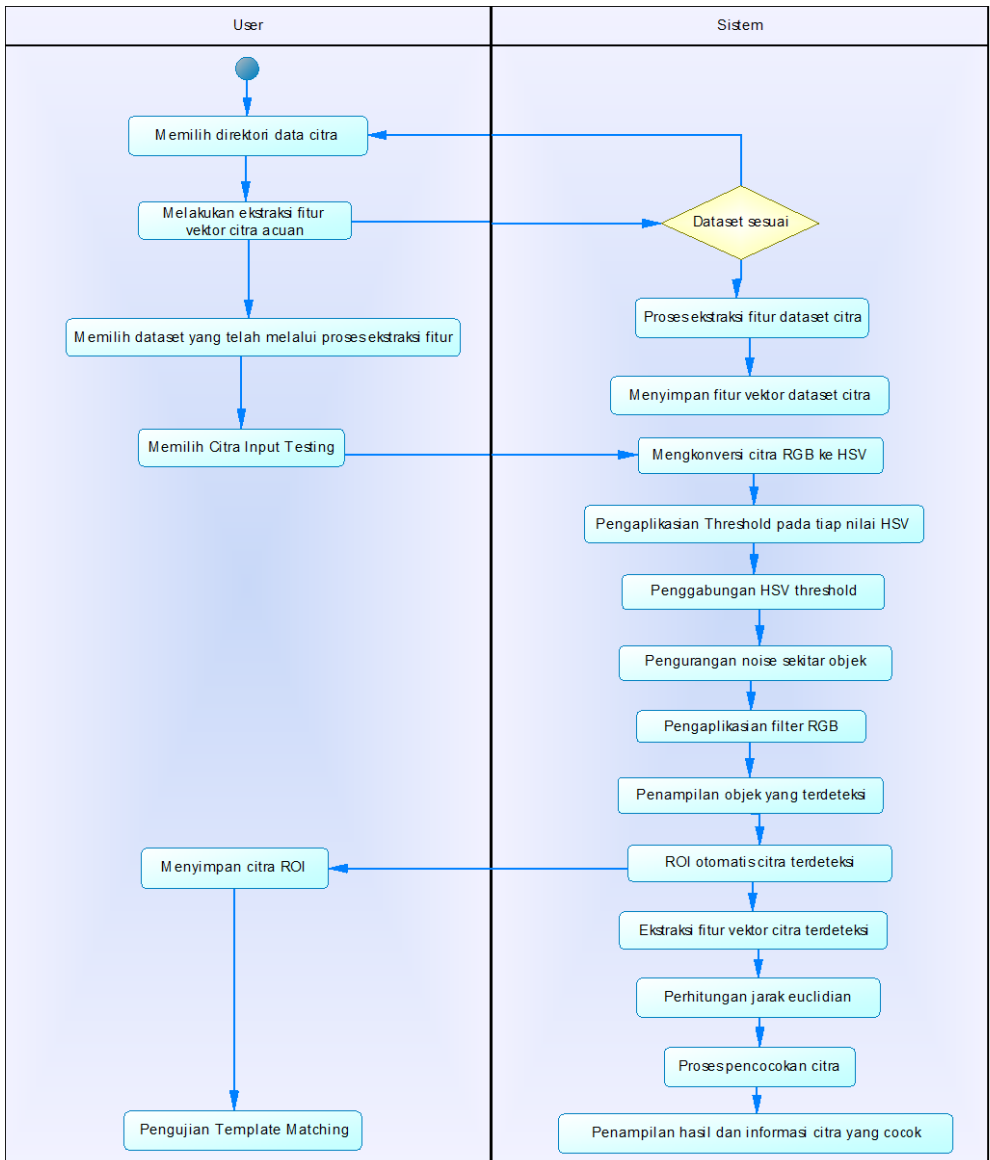
Dalam sub bab ini akan dijelaskan proses analisis dalam membangun perangkat lunak pengenalan rambu lalu lintas, mulai dari *Use Case Diagram*, *Activity Diagram*, proses pengambilan gambar, segmentasi citra, mendeteksi objek lalu lintas, mendeteksi objek dalam citra, proses pengenalan objek rambu lalu lintas dengan metode *Template Matching*, serta pengujian hasil dari sistem.

4.1.1 Analisis Sistem Perangkat Lunak

Perangkat lunak dibangun untuk menganalisis citra objek rambu lalu lintas yang dideteksi dengan segmentasi HSV dan dipilih dengan ROI dan melakukan pengenalan objek rambu lalu lintas tersebut. Secara Umum *Use Case Diagram* dan *Activity Diagram* disajikan pada Gambar 4.1 dan Gambar 4.2 menunjukkan jalannya sistem perangkat lunak.



Gambar 4.1. *Use Case Diagram* Pengenalan Rambu Lalu Lintas



Gambar 4.2. *Swimlane Diagram* Pengenalan Rambu Lalu Lintas

Sistem perangkat lunak yang dibangun ini memiliki beberapa tahapan sebagai berikut :

a. Akuisisi citra

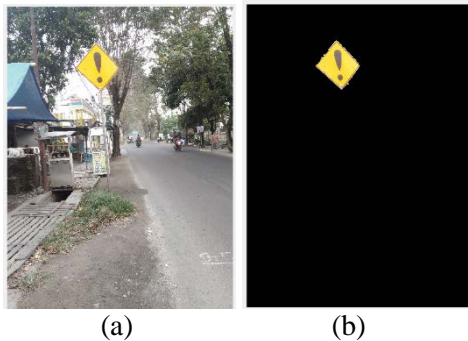
Akuisisi citra adalah tahapan awal untuk mendapatkan citra digital. Tujuannya adalah untuk menentukan data yang diperlukan dan memilih citra digital. Tahap ini dimulai dari objek yang akan diambil, persiapan alat-alat, sampai pada pencitraan. Pencitraan adalah kegiatan merubah dari citra tampak menjadi citra digital.

Data citra *input* tersebut harus memenuhi kriteria :

- Kamera dalam keadaan statis, artinya kamera berada dalam satu posisi selama pengambilan gambar.
- Citra input hanya memuat *single* objek.
- Hasil citra memiliki ukuran resolusi 3072x4096 pixel yang nantinya akan diubah menjadi resolusi 256x384 pixel untuk proses pengenalan.

b. Segmentasi citra

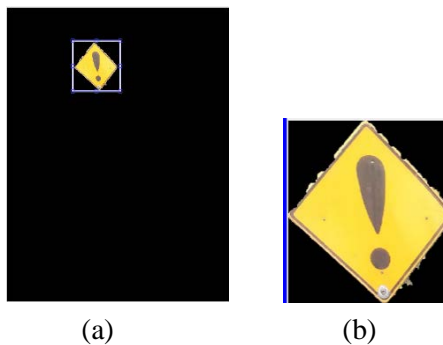
Segmentasi citra adalah proses pembagian suatu citra menjadi wilayah-wilayah yang homogen berdasarkan kriteria keserupaan yang tertentu antara tingkat warna HSV suatu piksel dengan tingkat warna HSV piksel-piksel tetangganya. Proses ini memiliki tujuan yang hampir sama dengan proses klasifikasi tidak terpadu. Dalam konteks citra digital daerah hasil segmentasi tersebut merupakan kelompok piksel yang bertetanggaan atau berhubungan. Dalam segmentasi ini, pendekatan daerah digunakan untuk membagi citra ke daerah-daerah sehingga didapatkan suatu daerah sesuai dengan kriteria yang diinginkan. Segmentasi mode warna HSV (*Hue, Saturation, Value*) digunakan untuk mendapatkan objek yang diinginkan. Kemudian hasil dari proses ini akan digunakan untuk proses lebih lanjut yaitu pemilihan ROI (*Region of Interest*).



Gambar 4.3. (a) Citra input (b) Citra hasil proses segmentasi HSV

c. Penentuan ROI (*Region of Interest*)

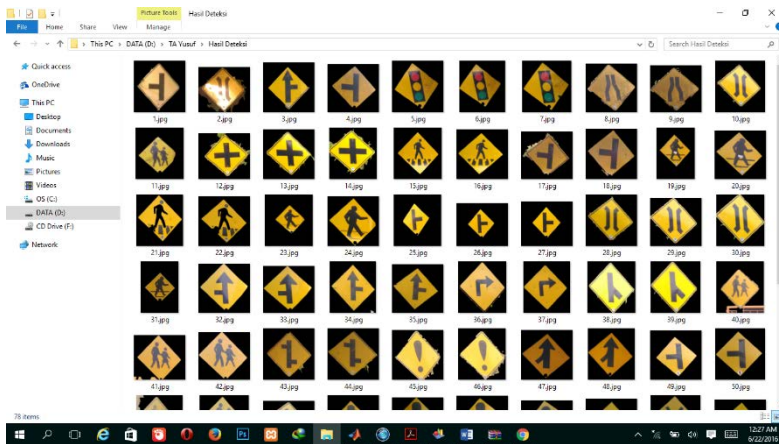
ROI (*Region of Interest*) digunakan sebagai area segmentasi atau pemilihan objek yang akan tersimpan sebagai dataset dan data citra uji pengenalan. Berdasarkan ROI ini user dapat memilih objek sesuai kebutuhan, dengan kata lain piksel-piksel selain dalam area ROI tidak akan terpilih.



Gambar 4.4 (a) Citra dengan ROI (daerah didalam kotak biru) (b) Citra hasil ROI

d. Proses penyimpanan objek rambu lalu lintas

Objek hasil proses deteksi akan disimpan dalam satu folder yang sama dalam direktori tertentu yang nantinya data tersebut akan diolah menjadi sebuah dataset. Dataset tersebut berupa ekstraksi fitur vektor yang disimpan dalam file dengan tipe *.mat yang nantinya fitur vektor tersebut akan dihitung jarak kemiripannya dengan fitur vektor citra uji menggunakan metode *template matching*.

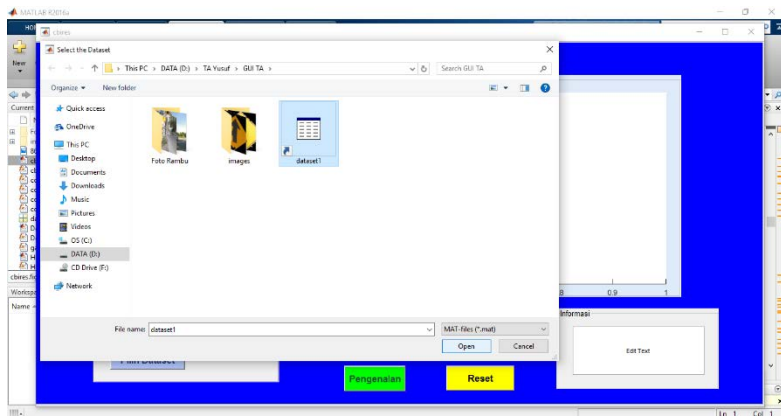


Gambar 4.5 Kumpulan data citra deteksi yang disimpan dalam direktori.

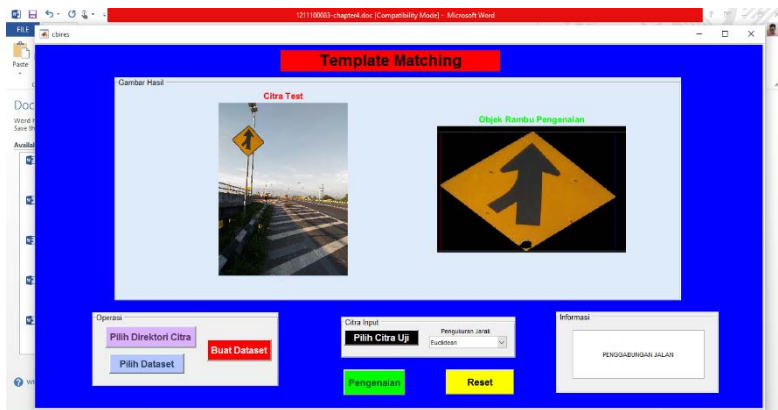
e. Proses *Template Matching*

Hasil deteksi menghasilkan objek rambu lalu lintas yang diinginkan. Setelah melewati tahapan tersebut maka objek akan diubah ukuran skalanya yang bertujuan untuk menyamakan ukuran citra uji dengan citra *template* dataset yang ada. Pada proses pencocokan *template matching* data citra uji tidak akan bisa diproses jika ukuran skalanya tidak sama. Proses selanjutnya yaitu citra uji dan citra *template* acuan akan di-ekstraksi fitur vektor dengan metode *color moment*. Setelah itu dilanjutkan ke proses *template*

matching dimana proses ini dilakukan pencocokan satu persatu fitur vektor data citra uji dengan fitur vector citra *template* database yang ada dengan menggunakan pengukuran jarak *euclidean*. Perhitungan jarak ini didapat dari pengurangan antara fitur vektor dataset citra *template* dengan fitur vektor data citra uji. Selisih dari pengukuran tersebut diharapkan mendekati nilai 0 (nol) atau nilai yang terkecil yang bisa dikatakan bahwa data citra uji tersebut “match” atau cocok. Setelah pencocokan akan ditampilkan hasil informasi dari objek rambu lalu lintas yang telah dikenali.



Gambar 4.6 Proses penyimpanan ciri fitur vektor citra, file citra ekstraksi disimpan dalam file *dataset.mat* yang nantinya akan dipanggil ketika melakukan proses *template matching*



Gambar 4.7 Proses *template matching* dan tampilan hasil pencocokan citra dan informasinya

4.1.2 Analisis Kebutuhan Sistem

Perangkat lunak ini dibangun menggunakan software MATLAB baik dari desain interface, sistem *toolbox computer vision* dan pengolahan citra digital.

Tabel 4.1 Tabel kebutuhan sistem

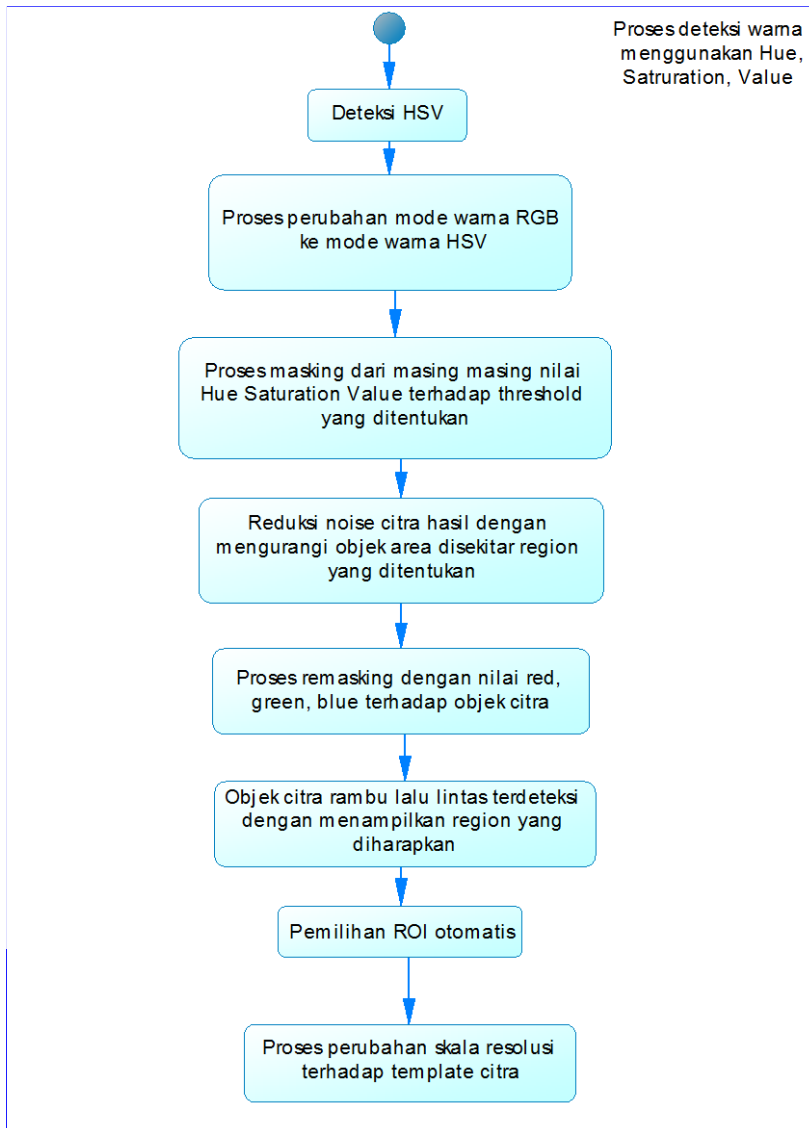
Perangkat Keras (Camera Digital)	Resolution	:13 Megapixel, f/2.0, 28mm, laser autofocus, dual-LED (dual tone) flash, 1/3" sensor-size
	Prosesor	: Quad-core Cortex-A53
	RAM	: 3 GB
Perangkat Keras (Personal Computer)	Prosesor	:Intel Core-i5@1.80Ghz(2.0 CPUs),~2.1Ghz
	RAM	: 4 GB
Perangkat Lunak	Sistem Operasi	: Windows 10
	Tools	: MATLAB R2016a

4.2 Perancangan Sistem

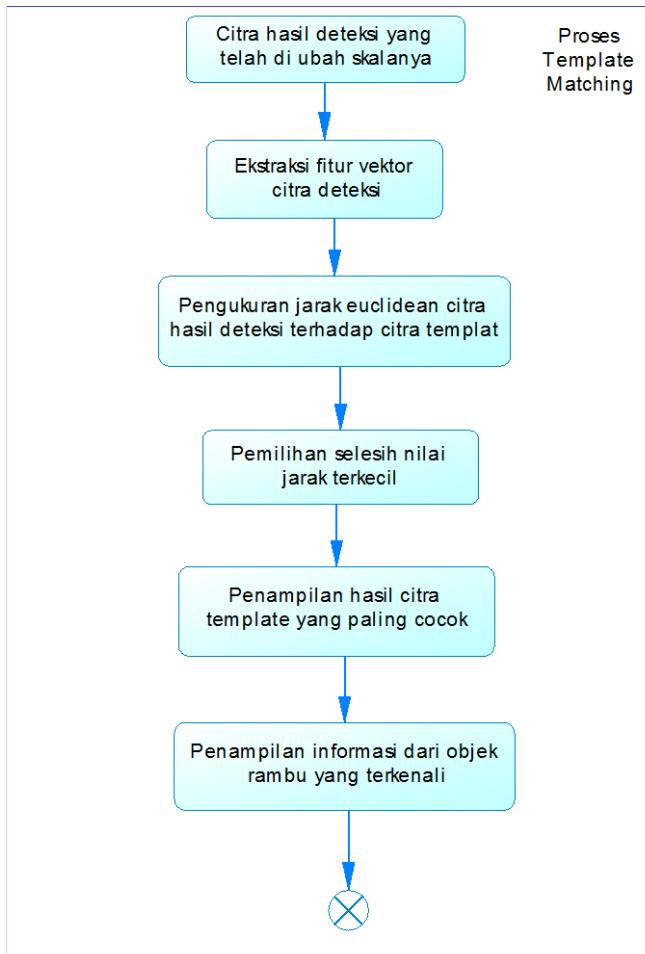
Setelah analisis sistem, kemudian dilanjutkan dengan perancangan sistem. Perancangan sistem tersebut meliputi perancangan data sistem, perancangan proses algoritma sistem, dan perancangan antarmuka sistem.

4.2.1 Gambaran Sistem

Pada subbab ini akan disajikan diagram alir sistem secara umum berupa proses berjalannya sistem secara umum yang terbagi menjadi 2 tahapan yaitu deteksi objek dengan metode deteksi *hue*, *saturation*, *value* lalu dilanjut pemilihan objek kandidat deteksi menggunakan ROI, dan pengenalan objek kandidat menggunakan metode *template matching*. Berikut diagram alir sistem yang disajikan pada gambar 4.8 dan gambar 4.9



Gambar 4.8 Diagram Alir Sistem Deteksi



Gambar 4.9 Diagram Alir Sistem Pengenalan

4.2.2 Penjelasan Umum Sistem

Proses algoritma ini dimulai dengan proses deteksi HSV (*Hue, Saturation, Value*) citra input. Didalam proses ini citra input yang mempunyai mode warna RGB akan diubah ke mode HSV (*Hue, Saturation, Value*). Setelah di dapatkan nilai dari masing-masing *Hue, Saturation*, dan *Value* maka proses selanjutnya akan didapatkan objek rambu yang diinginkan tetapi dalam proses ini masing-masing nilai HSV dikenai proses *masking* yang menyebabkan citra hasilnya menjadi berwarna hitam putih. Setelah terdeteksi objek rambu lalu lintas dalam citra tersebut, seringkali terdapat noise berupa area yang lebih kecil dari objek yang diinginkan yang ikut ditampilkan dalam satu frame objek terdeteksi. Maka setelah proses tersebut akan dilakukan pengurangan noise yaitu berupa pengurangan region kecil yang berada disekitaran objek deteksi yang diinginkan. Setelah itu akan diproses dengan penerapan *masking* mode warna *red, green*, dan *blue* terhadap citra tersebut untuk merubah ke mode warna RGB seperti semula. Lalu diteruskan ke proses ROI untuk mendapatkan daerah objek yang diinginkan lalu dilakukan proses pemotongan objek dalam satu frame citra. Setelah proses tersebut, citra kandidat terdeteksi diubah skala resolusinya yang bertujuan untuk mempermudah dalam proses pengenalan.

Pada proses pengenalan menggunakan metode *template matching*, dimana citra kandidat terdeteksi akan di ekstraksi fitur vektornya. Fitur vektor terdiri dari sekumpulan nilai yang nantinya fitur vektor citra kandidat terdeteksi akan dihitung dengan hasil ekstraksi fitur vektor citra *template* yang ada. Perhitungan jarak *euclidean* dalam hal ini digunakan untuk menghitung nilai paling minimal dari sesilih jarak antara citra uji dengan citra *template*. Selisih jarak yang terkecil nantinya ditampilkan oleh sistem sebagai citra yang dikenali.

4.2.2.1 Perancangan Data

Data yang digunakan dalam pengenalan rambu lalu lintas ini dibagi menjadi 3 macam, yaitu data masukan yang didapatkan dari

pengguna perangkat lunak, data proses yang dibutuhkan dan dihasilkan selama proses eksekusi perangkat lunak, dan data keluaran yang memberikan hasil proses eksekusi perangkat lunak untuk pengguna yang menjalankannya, yaitu citra terkenali sebagai rambu lalu lintas.

4.2.2.2 Data Masukan

Data masukan berupa citra digital hasil pemotretan yang berlokasi di jalan raya, citra tersebut berisi objek rambu lalu lintas yang akan diuji coba ke sistem. Dalam hal ini, data citra masukan merupakan hasil pemotretan langsung berlokasi di lingkup wilayah Kota Surabaya, Kota Mojokerto, dan Kota Jombang. Citra hasil pemotretan ini mempunyai resolusi default yang mengikuti kamera digitalnya, yaitu berukuran resolusi 3072x4096 piksel, mode warna RGB, format .jpg, dan dengan mode pemotretan auto.

4.2.2.3 Data Proses

Terdapat beberapa proses utama yang akan dilalui data citra yaitu deteksi segmentasi warna HSV, *Region of Interest*, proses merubah skala citra kandidat deteksi, ekstraksi fitur vektor *color moment*, proses pencocokan *template matching*.

4.2.2.4 Data Luaran

Proses paling akhir adalah pengujian pengenalan sistem. Data hasil dari proses tersebut adalah pengenalan objek rambu lalu lintas dan penampilan informasi serta jarak *euclidean*.

4.3 Penjelasan Proses Sistem

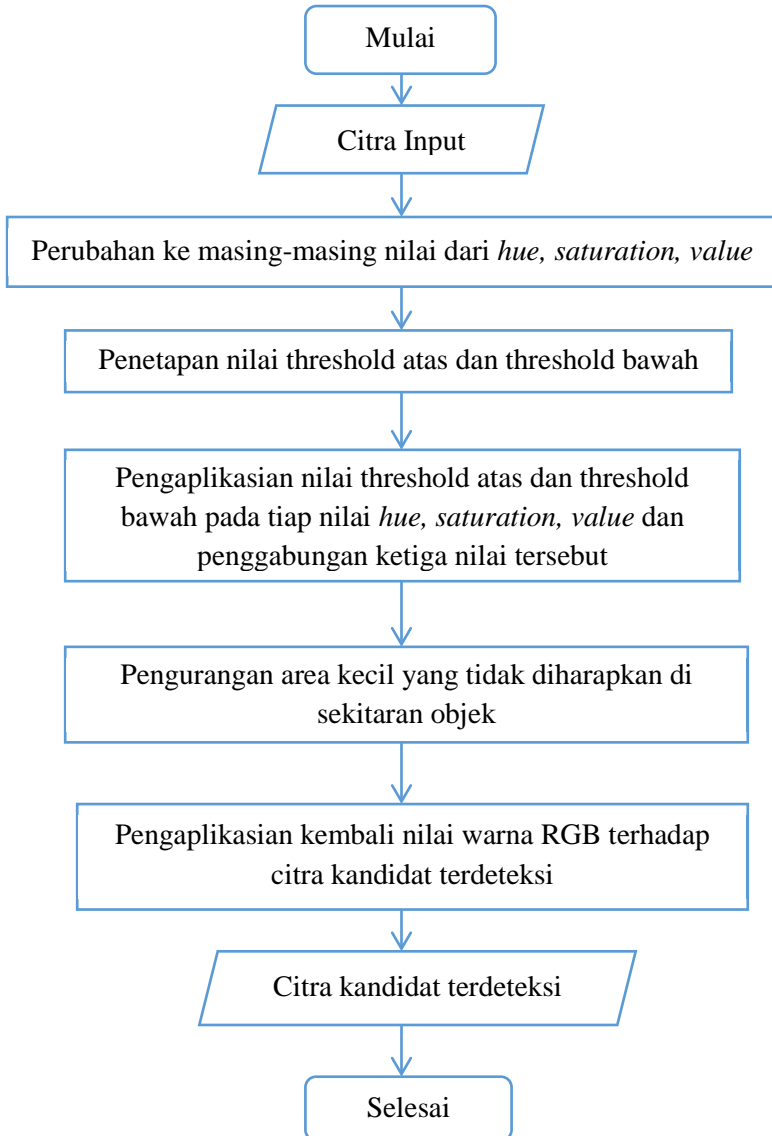
Pada subbab ini akan diuraikan lebih dalam tentang proses yang ada didalam sistem perangkat lunak. Dalam hal ini bertujuan untuk memperjelas hubungan antar proses beserta langkah-langkah yang dilakukan pada setiap proses. Secara garis besar proses yang dilakukan sebagai berikut :

1. Data input, berupa citra yang berisi single objek rambu lalu lintas.

2. Proses deteksi menggunakan segmentasi warna, adalah serangkaian proses untuk mendapatkan objek rambu lalu lintas yang diinginkan dengan segmentasi warna *hue*, *saturation*, *value*.
3. Proses ROI, merupakan lanjutan dari proses sebelumnya. Setelah kandidat citra uji terdeteksi maka akan dipilih daerah yang diinginkan untuk menghasilkan objek yang lebih sempurna dalam satu citra.
4. Proses perubahan skala berdasarkan template citra acuan, pada proses ini citra kandidat akan diubah skala resolusinya sesuai dengan *template* citra acuan agar pada proses perhitungan jarak dapat dengan mudah dilakukan.
5. Pembuatan dataset template citra acuan, adalah proses pembuatan dataset citra rambu lalu lintas terdeteksi yang diekstraksi dalam bentuk fitur vektor berupa *color moment*.
6. Ekstraksi fitur vektor citra uji terdeteksi, adalah proses ekstraksi fitur vektor dari citra kandidat terdeteksi.
7. *Template Matching*, proses ini merupakan pengukuran jarak kesamaan antara citra uji dengan citra *template* acuan. Perhitungan fitur vektor dalam hal ini menggunakan perhitungan jarak *euclidean*.

4.3.1 Proses Deteksi

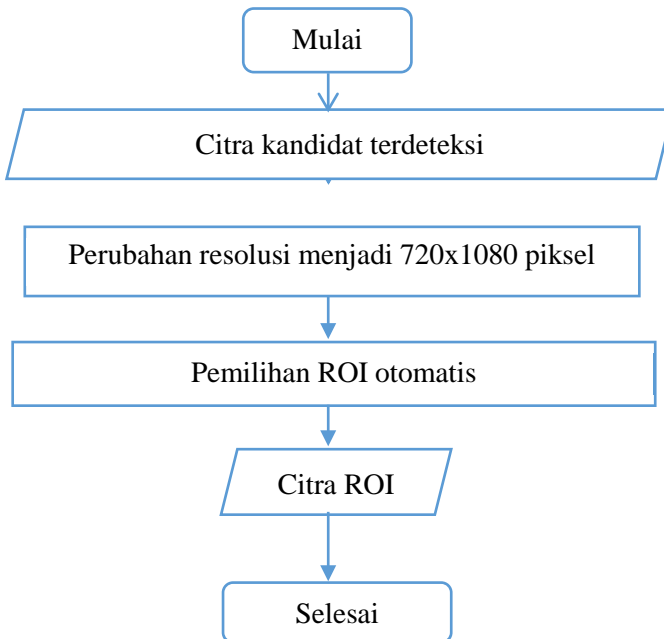
Proses ini merupakan tahap awal dalam sistem. Perubahan mode warna RGB ke HSV dilakukan pada tahap ini. Dari mulai mendapatkan nilai dari masing-masing *hue*, *saturation*, *value* sampai pengembalian ke mode warna RGB diterapkan untuk mengembalikan mode warnanya dengan tujuan warna dari objek rambu lalu lintas tidak berubah.



Gambar 4.10 Diagram Alir Proses Deteksi

4.3.2 *Region of Interest*

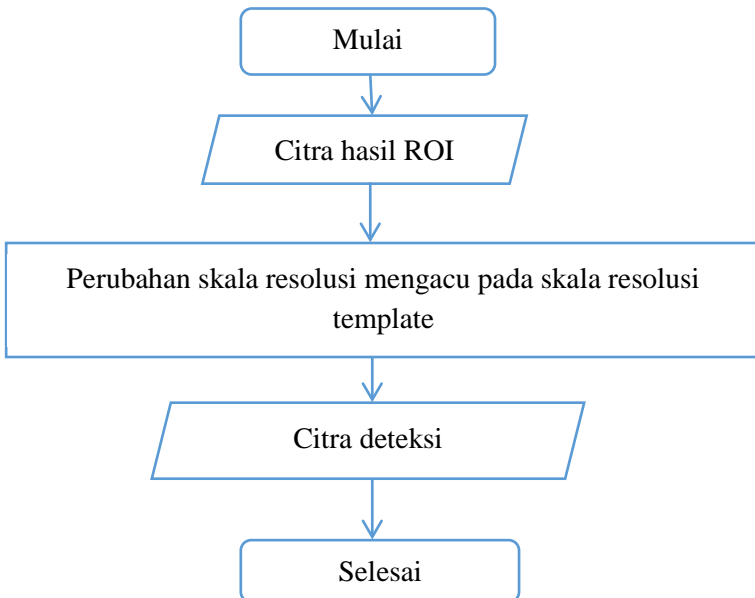
Setelah mendapatkan hasil keluaran dari proses pertama, maka citra tersebut diproses ke proses ROI, dimana proses ini akan didapatkan citra kandidat dengan wilayah yang diinginkan. Proses ini bertujuan untuk mendapatkan objek rambu lalu lintas lebih jelas dalam satu frame citra.



Gambar 4.11 Diagram Alir Proses ROI

4.3.3 Proses Perubahan Skala Resolusi

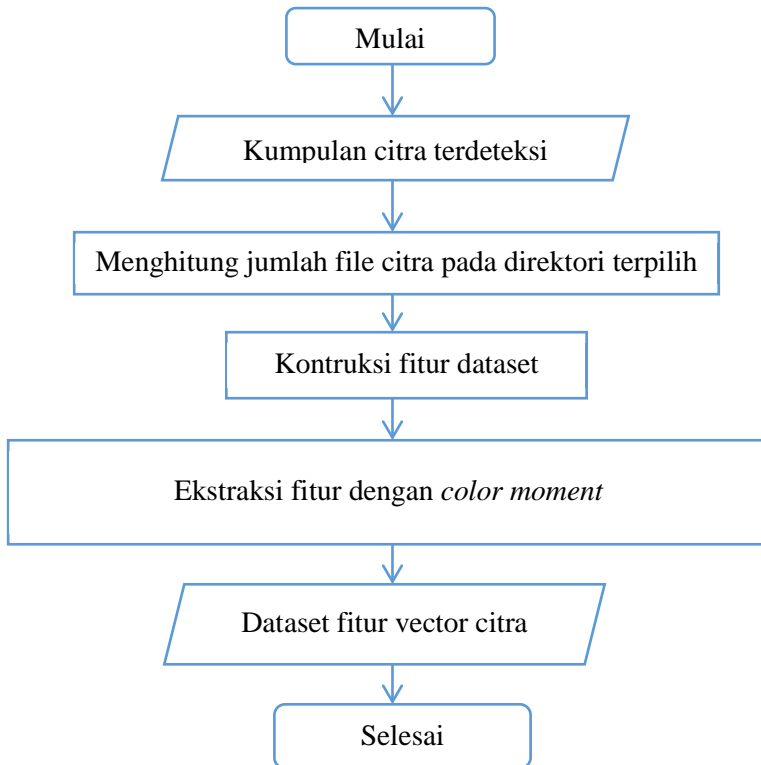
Proses perubahan skala resolusi ini dilakukan untuk mendapatkan skala yang sama dengan skala *template* yang juga untuk memudahkan proses pencocokan citra.



Gambar 4.12 Diagram Alir Proses *Rescalling*

4.3.4 Pembangunan Dataset

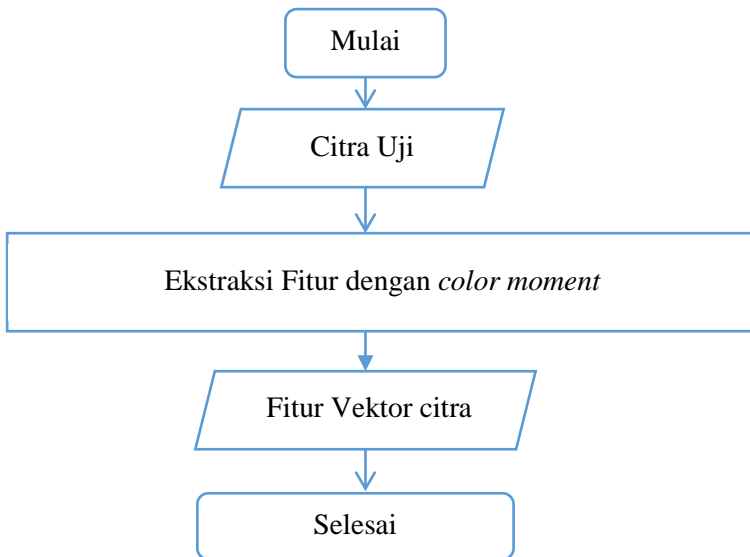
Pada proses ini akan dibuat sebuah dataset yang berisikan fitur vektor dari citra data hasil deteksi dan dataset yang berisi informasi dari masing-masing data tersebut. Dataset ini berfungsi sebagai data acuan untuk proses pencocokan citra uji yang akan diuji coba ke sistem.



Gambar 4.13 Diagram Alir Proses Pembuatan Dataset

4.3.5 Ekstraksi Fitur

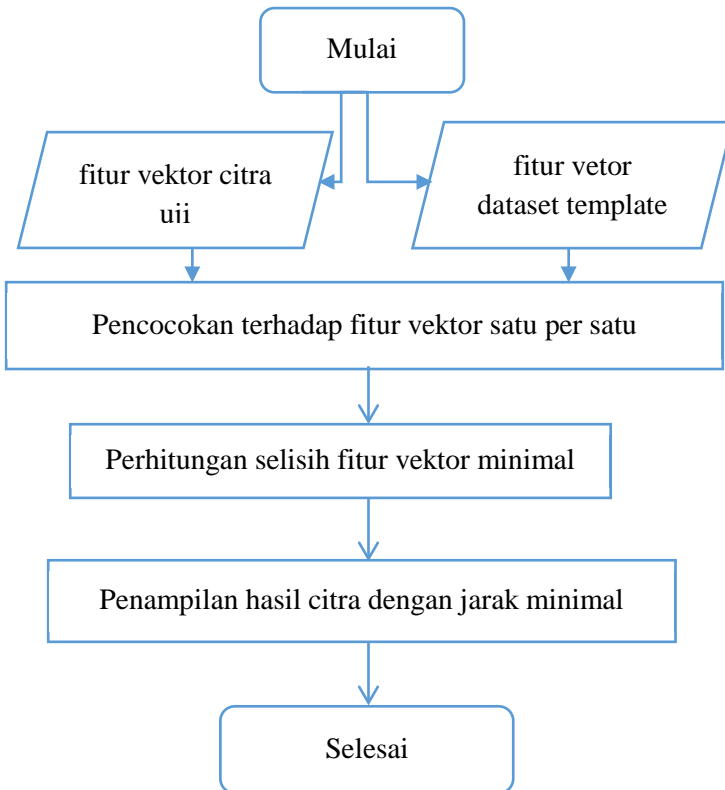
Ekstraksi fitur pada data citra kandidat terdeteksi dilakukan dengan mengambil fitur *color moment*. Setelah terambil ekstraksinya, fitur tersebut akan melakukan pencocokan dengan dataset fitur vektor yang telah dibuat dan mencari nilai jarak *euclidean* yang paling kecil.



Gambar 4.14 Diagram Alir Proses Ekstraksi Fitur

4.3.6 *Template Matching*

Proses *template matching* merupakan proses akhir dari system. Dimana di proses ini akan dilakukan proses pencocokan citra, antara citra kandidat dan citra template dataset.



Gambar 4.15 Diagram Alir Proses *Template Matching*

4.4 Implementasi Sistem

Perancangan program yang telah dibangun selanjutnya diimplementasikan pada bahasa pemrograman menggunakan *software* Matlab. Pembahasan dalam implementasi sistem meliputi

implementasi antarmuka (*interface*) system, implementasi tahap deteksi, implementasi tahap ekstraksi fitur dan pembuatan dataset, implementasi tahap pengenalan dengan metode *template matching*.

4.4.1 Implementasi Antarmuka

Pada tugas akhir ini, antarmuka system dibangun dengan menggunakan fitur form dan kontrol yang tersedia pada Matlab. Adapun antarmuka yang diimplementasikan pada tugas akhir ini adalah sebagai berikut.

4.4.1.1 Halaman Utama

Halaman Utama merupakan antarmuka yang berisi menu-menu untuk menampilkan antarmuka lainnya dalam system ini. Hasil implementasi bisa dilihat pada Gambar 4.16 berikut.



Gambar 4.16 Antarmuka Halaman Utama

Halaman utama dibuat dalam bentuk yang sederhana, terdiri dari 3 bagian utama, diantaranya :

1. Bagian *title bar*
2. Bagian *menu bar*
3. Bagian *main window*

Title bar merupakan bagian yang menunjukkan judul dari antarmuka yang sedang ditampilkan. *Main window* merupakan bagian antarmuka yang digunakan untuk menampilkan berbagai

antarmuka lain di dalam system. Dibawah *title bar* terdapat *menu bar* yang berisi sederetan menu-menu yang ditampilkan pada antarmuka utama sistem disajikan dalam Tabel 4.1 berikut ini :

Tabel 4.2 Tabel Kegunaan Menu Sistem

Menu	Kegunaan
Deteksi	Melakukan deteksi citra yang berisi objek rambu lalu lintas yang bertujuan mengetahui apakah citra tersebut dapat dikenali sistem atau tidak melalui memasukkan citra secara manual.
Pengenalan	Melakukan pembuatan dataset template citra acuan dengan cara meng-ekstraksi fiturnya dan menyimpan dataset dalam bentuk *.mat serta melakukan pengujian, pengamatan, dan penilaian pada sistem, seberapa baik hasil citra kandidat terdeteksi diuji coba pengenalan menggunakan metode <i>template matching</i> .
Keluar	Keluar dari sistem

4.4.1.2 Antarmuka Deteksi

Antarmuka deteksi berguna untuk melakukan pengujian deteksi terhadap citra yang memuat objek rambu lalu lintas atau tidak. Antarmuka deteksi ini juga bertujuan untuk mengamati objek rambu lalu lintas apa saja yang dapat terdeteksi menggunakan system ini. Pada tampilan ini juga akan dilakukan pemilihan ROI sehingga pengguna dapat memilih daerah ROI dari objek yang diinginkan. Hasil implementasi antarmuka deteksi dapat dilihat pada gambar 4.17 berikut.



Gambar 4.17 Antarmuka Deteksi

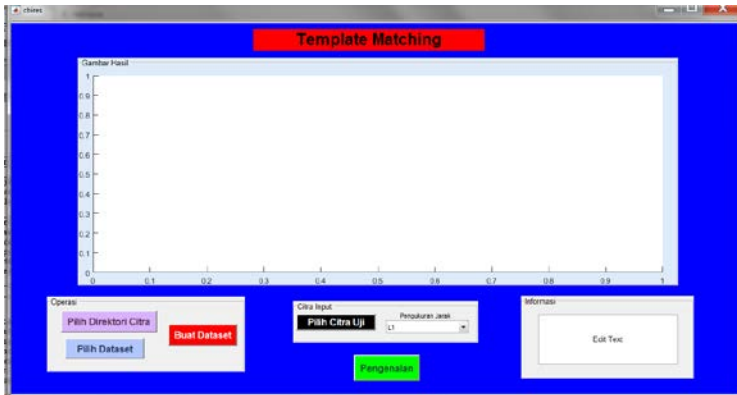
Pada antarmuka deteksi terdapat 5 buah tombol yang memiliki fungsi yang berbeda-beda. Fungsi dari tombol tersebut adalah sebagai berikut :

1. Tombol pilih gambar digunakan untuk mengambil citra rambu lalu lintas dari direktori tertentu yang tersimpan pada komputer.
2. Tombol Segmentasi HSV untuk melakukan segmentasi warna pada citra masukkan dan menghasilkan segmentasi warna dari objek yang diinginkan.
3. Tombol ROI untuk memilih daerah sekitaran objek yang diinginkan.
4. Tombol *Resize* untuk mengubah skala citra dari ukuran citra asli ke ukuran skala citra template acuan agar mempermudah pada saat proses *template matching*.
5. Tombol *reset* untuk mengembalikan ke tampilan kosong awal semula.

4.4.1.3 Antarmuka Pengenalan (*Template Matching*)

Antarmuka pengenalan memiliki fungsi untuk melakukan 2 hal. Pertama, pada antarmuka ini terdapat tombol, 'Pilih Direktori Citra', 'Buat Dataset', dan 'Pilih Dataset' yang berfungsi untuk membangun sebuah dataset fitur vektor citra template acuan dan menyimpan dataset tersebut dalam direktori komputer dan memilih dataset saat diperlukan untuk proses uji coba *template matching*. Kedua, pada antarmuka ini berfungsi untuk melakukan proses uji

coba citra kandidat yang terdeteksi dari proses sebelumnya dan diteruskan ke proses pengenalan dengan metode *template matching* dan hasil citra yang cocok akan ditampilkan beserta informasi dan nilai jarak kesamaannya terhadap citra template acuan. Hasil implementasi antarmuka pengenalan dapat dilihat pada gambar 4.18 berikut.



Gambar 4.18 Antarmuka Pengenalan

Pada antarmuka pengenalan terdapat 5 tombol, 1 *popup menu*, 1 *edit text box*. Fungsi dari tombol-tombol tersebut adalah sebagai berikut.

1. Tombol Pilih Direktori Citra digunakan untuk memilih folder yang berisi citra yang akan diproses untuk uji coba pengenalan dan dijadikan menjadi satu dataset.
2. Tombol Buat Dataset digunakan untuk mengekstraksi fitur vektor dari citra yang sudah dipilih dan citra yang akan di uji coba dalam proses pengenalan. Hasil dari tombol ini adalah sebuah file dataset yang mempunyai format *.mat dan disimpan ke direktori tertentu di komputer.
3. Tombol Pilih Dataset digunakan untuk memilih dataset yang telah dibuat dan yang telah tersimpan di komputer untuk proses pengujian *template matching*.
4. Tombol Pilih Citra Uji untuk mengambil citra uji dari citra kandidat yang telah melewati proses deteksi.

5. *Popup menu* Pengukuran Jarak untuk memilih perhitungan kemiripan antara 2 citra menggunakan perhitungan jarak *Euclidean* atau *Normalized Euclidean*.
6. Tombol pengenalan untuk mengeksekusi proses pengujian pencocokan 2 citra dan menampilkan hasil dari proses tersebut.
7. *Edit text box* berfungsi untuk menampilkan informasi dari rambu lalu lintas.

4.4.2 Implementasi Tahap Input Data

Proses input data membutuhkan interaksi pengguna untuk mencari dan mengambil citra yang akan diuji coba. Proses ini diimplementasikan pada sebuah program sebagai berikut :

```
[FileName, PathName, FilterIndex] =
uigetfile('*.jpg; *.png', 'Pilih Citra',
'*.jpg');
handles>NamaFile = FileName;
A.FileName = FileName;
handles.PathName = PathName;
A.PathName = PathName;
guidata(hObject,handles);
axes(handles.axes1);
handles.Citra.RGB = imread([PathName
FileName]);
A.Image = handles.Citra.RGB;
imshow(A.Image);
```

4.4.3 Impelemntasi Proses Segmentasi HSV

Pada proses segmentasi ini, dilakukan pengubahan citra dari citra RGB ke HSV dan citra biner, dengan nilai *threshold* yang telah ditentukan sehingga menghasilkan citra yang bisa membedakan daerah rambu dan daerah latar tempat dengan baik. Kemudian dilakukan pengembalian ke citra RGB kembali dengan melakukan *masking* warna RGB ke citra biner untuk menghasilkan citra yang diinginkan. Proses ini diimplementasikan pada sebuah program sebagai berikut :

```

rgbImage = A.Image;
hsvImage = rgb2hsv(rgbImage);
% Extract out the H, S, and V
hImage = hsvImage(:,:,1);
sImage = hsvImage(:,:,2);
vImage = hsvImage(:,:,3);
hueThresholdLow = 0;
hueThresholdHigh = graythresh(hImage);
saturationThresholdLow = graythresh(sImage);
saturationThresholdHigh = 1.0;
valueThresholdLow = graythresh(vImage);
valueThresholdHigh = 1.0;
saturationThresholdHigh = 1.0;
valueThresholdLow = graythresh(vImage);
valueThresholdHigh = 1.0; maskedImageR =
coloredObjectsMask .* rgbImage(:,:,1);
maskedImageG = coloredObjectsMask .*
rgbImage(:,:,2);
maskedImageB = coloredObjectsMask .*
rgbImage(:,:,3);

```

4.4.4 Implementasi Proses ROI

Pada proses ROI citra yang telah terdeteksi melalui proses segmentasi HSV akan diambil daerah persekitaran objek yang dipilih. Diharapkan melalui proses ini objek dapat dipilih dan memudahkan untuk dikenali pada proses selanjutnya. Implementasi programnya adalah sebagai berikut :

```

I= maskedRGBImage;
I=imresize(I,[720 480]);
[m n] = size(I);
k1 = -1;
k2 = -1;
a = 0;
for i=1:m
    for j=1:n
        if (I(i,j)>0 && I(i,j)<255)
            k1 = i;
            k2 = j;

```

Proses ROI ini menggunakan fungsi yang tersedia pada matlab untuk merubah skala resolusi citra uji ke skala resolusi citra template acuan. Implementasi programnya adalah sebagai berikut :

```
crop = imcrop(I);
crop2 = imcrop(crop);
si = imresize(crop2,[384 256]);
set(imshow(crop2));
```

4.4.6 Implementasi Pembangunan *Dataset*

Pembangunan dataset adalah sebuah proses ekstraksi fitur dari sejumlah citra data yang akan dijadikan citra template acuan yang nantinya menjadi patokan penghitungan jarak antara citra template acuan dengan citra uji. Disini yang dihitung adalah fitur vektornya. Fitur vektor ini didapatkan dari perhitungan metode *color moment*. Implementasi programnya adalah sebagai berikut :

```
if (~isfield(handles, 'folder_name'))
    errordlg('Pilih folder citra terlebih
dahulu!');
    return;
end
jpgImagesDir = fullfile(handles.folder_name,
'*.jpg');
num_of_jpg_images = numel( dir(jpgImagesDir) );
jpg_files = dir(jpgImagesDir);
for k = 1:totalImages

    nama_file = '';
    if ( (num_of_jpg_images - jpg_counter)
> 0)
        imgInfoJPG = imfinfo( fullfile(
handles.folder_name,
jpg_files(jpg_counter+1).name ) );
        if ( strcmp(
lower(imgInfoJPG.Format), 'jpg') == 1 )
```


Setelah proses pembangunan dataset, maka dataset tersebut akan disimpan dalam direktori komputer, yang nantinya file dataset tersebut akan dipanggil ketika melakukan proses pengenalan. Implementasi programnya adalah sebagai berikut :

```
[fname, pthname] = uigetfile('*.mat', 'Select
the Dataset');
[fname, pthname] = uigetfile('*.mat', 'Select
the File Name');
```

4.4.8 Implementasi Pemilihan Citra Data Uji

Pemilihan citra uji akan dilakukan pengambilan kembali data citra yang telah melalui proses deteksi. Berikut implementasi programnya :

```
global S;
if isstruct(S) && isfield(S, 'Image')
queryImage = S.Image;
queryImage = imresize(queryImage, [384 256]);
else
[query_fname, query_pathname] =
uigetfile('*.jpg; *.png; *.bmp', 'Select query
image');
queryImage = imread( fullfile( pathstr,
strcat(name, ext) ) );
```

4.4.9 Implementasi Ekstraksi Fitur Citra Uji

Pada tahap ini akan di jelaskan ekstraksi fitur dari setiap data citra uji. Tahap ini akan mengubah citra yang bentuknya direpresentasikan dengan sebuah matriks [m n] akan diekstrak menjadi matrik [1 n] untuk tiap citranya. Proses pengubahanya menggunakan *color moment* sehingga nilai-nilai tersebut akan di simpan di dalam array [1 n]. Lalu array tersebut akan dilakukan proses pencocokan pada tahapan berikutnya. Berikut adalah implementasi programnya :

```
%ekstraksi fitur vektor
queryImage = imresize(queryImage, [384 256]);
Colormoment = colormoment(queryImage);
queryImageFeature = [colormoment];
handles.queryImageFeature = queryImageFeature;
S.Image = queryImage;
guidata(hObject, handles);
```

4.4.10 Implementasi Pemilihan Pengukuran Jarak

Proses perhitungan jarak pada sistem ini menggunakan perhitungan kesamaan jarak. Jarak yang digunakan adalah jarak *euclidean* dan *normalized euclidean*. Masing-masing matrik ekstrak fitur citra uji akan dihitung jaraknya dengan matrik fitur citra template acuan yang telah dijadikan dataset. Berikut adalah rumus dari jarak :

Jarak *Euclidean*: $d(\bar{x}, \bar{y}) = \sqrt{\sum_{i=1}^n (\bar{x}_i - \bar{y}_i)^2}$

Jarak *Normilized Euclidean* : $d(\bar{x}, \bar{y}) = \sqrt{\sum_{i=1}^n \frac{(\bar{x}_i - \bar{y}_i)^2}{\sigma}}$

Berikut adalah implementasi programnya :

```
euclidean = zeros(size(dataset, 1), 1);
dataset(:, end) = [];
euclidean = zeros(size(dataset, 1), 1);
for k = 1:size(dataset, 1)
    euclidean(k) = sqrt( sum( power(
dataset(k, :) - queryImageFeatureVector, 2 ) )
);
end
```

4.4.11 Implementasi Proses Eksekusi Pengenalan

Tahapan ini adalah proses pengenalan citra uji apakah citra tersebut dapat dikenali sebagai citra rambu lalu lintas atau tidak. Menggunakan metode *template matching* yang mana metode ini akan mengukur kesamaan jarak antara citra uji dengan citra template dan menampilkan hasil citra yang cocok. Dalam hal ini citra yang ditampilkan sebagai hasil adalah citra yang mempunyai jarak minimal. Berikut adalah implementasi programnya :

```

if (~isfield(handles, 'queryImageFeature'))
    errordlg('Pilih gambar test terlebih
dahulu');
    return;
end

if(~isfield(handles, 'folder_name'))
    errordlg('Pilih image directory terlebih
dahulu !');
    return;
end
if (~isfield(handles, 'imageDataset'))
    errordlg('Pilih dataset terlebih dahulu');
    return;
end

if ~isfield(S, 'Image')
    errordlg('Pilih gambar lagi');
    return;
end
if (~isfield(handles, 'DistanceFunctions') &&
~isfield(handles, 'numOfReturnedImages'))
    metric =
get(handles.popupmenu_DistanceFunctions,
'Value');
numOfReturnedImgs =
get(handles.popupmenu_NumOfReturnedImages,
'Value');
elseif (~isfield(handles, 'DistanceFunctions')
|| ~isfield(handles, 'numOfReturnedImages'))
    if (~isfield(handles,
'DistanceFunctions'))

```

4.4.12 Implementasi Penampilan Informasi

Informasi dari sebuah rambu lalu lintas sangat perlu untuk ditampilkan dikarenakan setiap symbol dari rambu lalu lintas akan mempunyai arti sendiri-sendiri. Berikut adalah implementasi programnya :

```
img_name = nama_file.n_file{indx(m)};  
set(edit3 , 'string', [img_name(1:end-4)  
char(13) ' Jarak : ' num2str(sortedDist(m))]);  
str_name = strcat(folder_name, '/', img_name);  
returnedImage = imread(str_name);
```

BAB V

PENGUJIAN DAN PEMBAHASAN

Pada bab ini akan dilakukan pengujian serta pembahasan terhadap program yang telah dibuat. Pengujian sistem ini merupakan pengujian deteksi dan pengenalan terhadap citra uji yang memuat objek rambu lalu lintas. Data tersebut didapat langsung dari pemotretan jalan raya di beberapa tempat lokasi. Data tersebut akan diuji coba pada sistem ini menggunakan metode pengenalan *template matching*. Hasil pengujian kemudian dibahas untuk mengetahui unjuk kerja sistem secara keseluruhan dalam menjalankan fungsi yang diinginkan. Pada akhir pada bab ini akan dilakukan pengukuran performa *template matching*.

5.1 Data Uji Coba

Uji coba pada program dalam penelitian tugas akhir ini dilakukan terhadap video berekstensi .jpg berjumlah 121 data citra. Dari hasil pengambilan data didapatkan data citra objek rambu lalu lintas yang terbanyak adalah rambu peringatan karena frekuensi kemunculan di jalan raya lebih sering dibandingkan objek rambu lalu lintas lainnya.





Gambar 5.1 Contoh beberapa data citra masukan.

5.2 Pengujian Tahap Deteksi dengan Segmentasi *Hue, Saturation, Value*

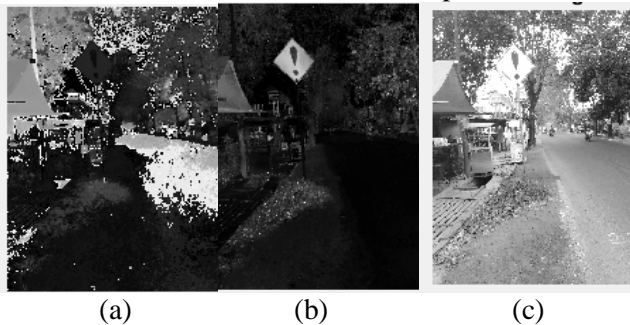
Pengujian ini dilakukan terhadap proses-proses pada tahap pengolahan citra. Pengujian bertujuan untuk mengetahui bahwa proses-proses pada tahap pengolahan citra sudah benar. Pada tahap ini terdapat 6 proses yaitu : konversi ke HSV, HSV *masking*, pereduksian *smallest area* ada di sekitar objek, *border smoothing*, RGB *masking*, penggabungan gambar asli dengan RGB *masking*.

1. Pengujian konversi ke mode warna HSV
Pengujian proses ini bertujuan untuk mengetahui bahwa sistem telah berhasil merubah menjadi mode warna HSV.
2. Pengujian HSV *masking*
Pengujian proses ini bertujuan untuk mengetahui bahwa sistem telah berhasil merubah menerapkan HSV *masking* pada masing nilai *Hue, Saturation, Value*.
3. Pengujian reduksi *smallest area*
Pengujian proses ini bertujuan untuk mengetahui bahwa sistem telah berhasil mereduksi noise yang berupa area kecil yang tidak diharapkan.

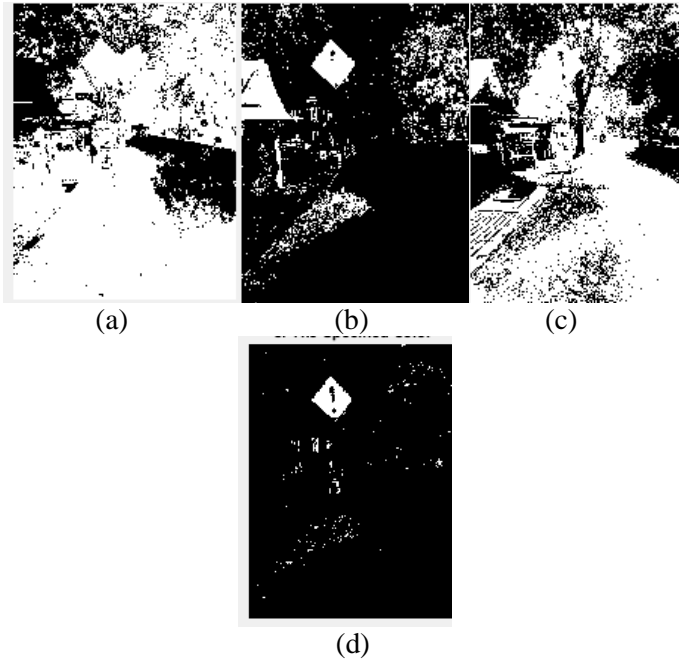
4. Pengujian *border smoothing*
Pengujian proses ini bertujuan untuk mengetahui bahwa sistem telah berhasil menghaluskan daerah *border*.
5. Pengujian *RGB masking*
Pengujian proses ini bertujuan untuk mengetahui bahwa sistem telah berhasil melakukan *RGB masking*.
6. Pengujian penggabungan citra asli dengan *RGB masking*
Pengujian proses ini bertujuan untuk mengetahui bahwa sistem telah berhasil menggabungkan citra asli dengan *RGB masking* dan menghasilkan segmentasi warna yang diharapkan.



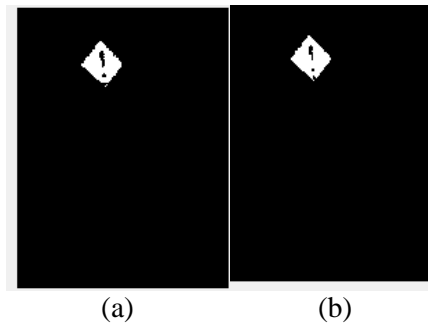
Gambar 5.2 Contoh data citra input



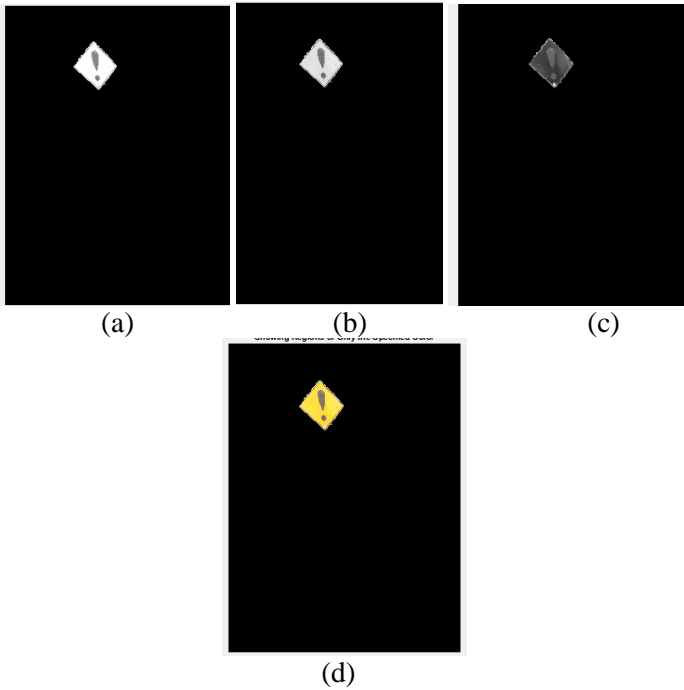
Gambar 5.3 (a)Citra *Hue*, (b)Citra *Saturation*, (c)Citra *Value*



Gambar 5.4 (a)Citra *Hue Masked*, (b)Citra *Saturation Masked*, (c)Citra *Value Masked*, (d)Citra hasil penggabungan ketiga citra



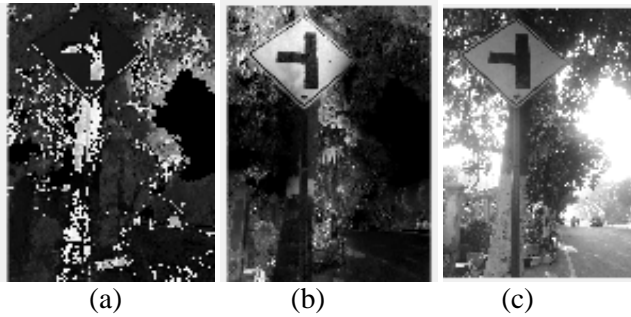
Gambar 5.5 (a)Citra reduksi *smallest area*, (b)Citra *border smoothing*



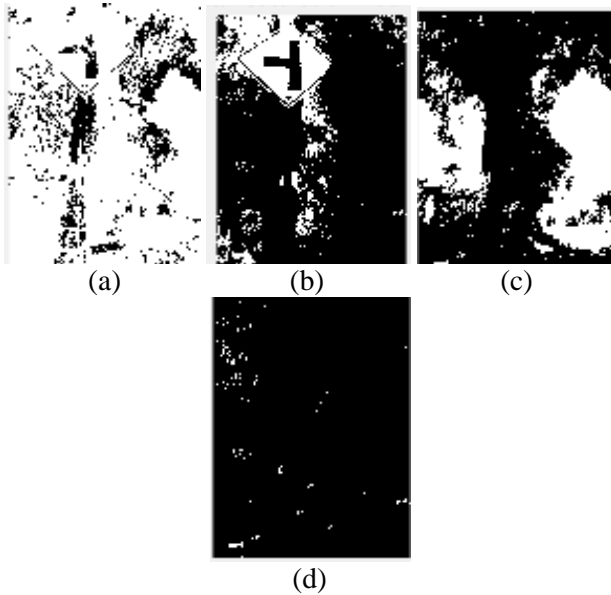
Gambar 5.6 (a)Citra *red masking*, (b)Citra *green masking*, (c)Citra *Blue masking*, (d)Citra hasil penggabungan ketiganya



Gambar 5.7 Contoh data citra input yang gagal terdeteksi



Gambar 5.8 (a)Citra *Hue*, (b)*Saturation*, (c)*Value*



Gambar 5.9 (a)Citra *Hue Masked*, (b)Citra *Saturation Masked*, (c)Citra *Value Masked*, (d)Citra hasil penggabungan ketiga citra yang menyebabkan gagal terdeteksi



Gambar 5.10 Citra hasil penggabungan *red, green, blue* channel menyebabkan hasil gambar hitam dan objek gagal terdeteksi

5.3 Pembahasan Konversi Mode Warna RGB ke HSV

Mode warna yang digunakan pada tahap deteksi adalah mode warna *Hue, Saturation, Value* (HSV) sedangkan citra input uji coba bermode warna *Red, Green, Blue* (RGB). Dalam hal ini perlu dihitung nilai untuk masing-masing komponen H, S, dan V. Nilai RGB menggunakan skala 0 sampai 255, sedangkan HSV menggunakan skala 0 sampai 1.

Misalkan untuk warna  memiliki nilai RGB sebagai berikut :

$$R = 52, G = 101, B = 164$$

Didapat nilai min adalah 52 dan max adalah 164. Dengan rumus berikut akan dicari nilai HSV nya :

$$H = \begin{cases} \text{Tidak terdefinisi, jika } \max = \min \\ \frac{60^\circ \times \frac{G-B}{\max-\min} + 0^\circ}{360}, \text{ jika } \max = R \text{ dan } G \geq B \\ \frac{60^\circ \times \frac{G-B}{\max-\min} + 360^\circ}{360}, \text{ jika } \max = R \text{ dan } G < B \\ \frac{60^\circ \times \frac{G-B}{\max-\min} + 120^\circ}{360}, \text{ jika } \max = G \\ \frac{60^\circ \times \frac{G-B}{\max-\min} + 240^\circ}{360}, \text{ jika } \max = B \end{cases}$$

$$S = \begin{cases} 0, \text{ jika } \max = 0 \\ 1 - \left(\frac{\min}{\max}\right), \text{ untuk lainnya} \end{cases}$$

$$V = \frac{max}{255}$$

$$H = \frac{60^\circ \times \frac{101 - 164}{164 - 52} + 240^\circ}{360} = \frac{206.25}{360} = 0.5729$$

$$S = 1 - \left(\frac{52}{164} \right) = 0.6829$$

$$V = \frac{164}{255} = 0.6431$$

Didapat nilai HSV dari warna diatas adalah H=0.5729, S=0.6829, dan V=0.6431. Contoh perhitungan tersebut hanyalah untuk permisalan warna diatas. Untuk warna lain akan mempunyai nilai RGB yang berbeda dan tentunya juga mempunyai konversi nilai HSV yang berbeda pula.

5.4 Pengujian Tahap Pemilihan ROI

Proses ini bertujuan untuk mengetahui bahwa sistem perangkat lunak berhasil memperoleh citra daerah sekitaran objek yang akan dikenali sebagai citra template acuan dan citra uji. Citra yang telah melewati proses ROI dapat dilihat pada Gambar 5.7 berikut ini,



Gambar 5.11 Citra hasil proses ROI berukuran 720x480 piksel

5.5 Pengujian Tahap Pengubahan Skala Resolusi

Proses ini bertujuan untuk merubah skala citra input terhadap citra template acuan. Pada proses kali ini citra input akan diubah menjadi ukuran 384x256 piksel sama seperti ukuran citra

template acuan. Berikut citra hasilnya dapat dilihat pada gambar 5.8 dibawah ini,



Gambar 5.12 Citra hasil proses perubahan resolusi berukuran 384x256 piksel

5.6 Pengujian Tahap Ekstraksi Fitur

Pengujian proses ini menggunakan data citra yang telah diproses pada proses deteksi. Tahap ekstraksi fitur ini menggunakan metode *color moment* yaitu nilai *mean red*, *std red*, *skw red*, *mean green*, *std green*, *skw green*, *mean blue*, *std blue*, *skw blue*.

Data pengujian ekstraksi fitur dengan citra input sampel dapat dilihat pada table 5.1 berikut :

Tabel 5.1 Hasil proses ekstraksi fitur dengan *color moment*

Ekstraksi Fitur		Citra Input
Color moment	Mean Red	83.5786234537761
	Std Red	86.8109293569521
	Skw Red	0.288881283947920
	Mean Green	69.8895772298177
	Std Green	71.6257958042379
	Skw Green	0.267299550621805
	Mean Blue	38.6995340983073
	Std Blue	44.2497161081435
	Skw Blue	0.964220789380644

5.7 Pengujian Tahap Pembangunan Dataset

Pengujian proses ini menggunakan 121 data citra yang berhasil melewati proses tahapan pengenalan yang dijadikan sebagai citra template acuan untuk proses pencocokan. Data citra template acuan tersebut secara keseluruhan akan diambil ekstrak fiturnya dan akan disimpan menjadi satu file dataset.mat dan nama file dari masing-masing citra tersebut juga akan diekstraksi dalam file lain yang isinya hanyalah string dari masing-masing citra tersebut.

Data luaran yang dihasilkan akan disimpan dalam file matlab dengan format *.mat. Data yang disimpan akan berbentuk informasi seperti berikut :

Citra	Color Moment								
Ke-	Mean R	Std R	Skw R	Mean G	Std G	Skw G	Mean B	Std B	Skw B
1	83.5786	86.8109	0.28888	69.8896	71.6258	0.2673	38.6995	44.2497	0.96422
2	70.8274	84.0464	0.39313	64.2432	76.5433	0.40569	47.4846	57.3586	0.4998
3	99.4345	102.189	0.25244	80.2508	82.9114	0.27263	19.691	32.6235	2.0173
4	106.847	113.71	0.28241	88.9655	93.399	0.2382	52.1685	56.1874	0.46784
5	97.2747	112.392	0.46562	75.0438	86.5987	0.46292	9.66046	19.1022	2.75644
6	84.4379	88.9818	0.24213	69.0926	72.1365	0.21999	13.6904	18.6723	1.79759
7	85.4741	94.2854	0.43106	67.9103	72.4009	0.35715	27.0014	28.2038	0.52324

Tabel 5.2 Hasil proses pembangunan dataset dan ekstraksi fitur dengan *color moment*

5.8 Pengujian Tahap Pengenalan

Pengujian proses ini ada 2 tahapan, yaitu pertama tahap proses deteksi yang sebelumnya dijelaskan di 5.1 dan kedua tahap pengenalan. Pada tahap deteksi data yang digunakan berjumlah 121 citra dan pada tahap pengenalan data yang digunakan berjumlah 121 citra yang telah berhasil melewati tahap pertama. Hasil pengujian sistem dapat dilihat pada tabel 5.3

Tabel 5.3 Hasil proses pengujian deteksi dan pengenalan

No	Data Citra	Deteksi	Pengenalan
1	1.jpg	V	V
2	2.jpg	V	V
3	3.jpg	V	V
4	4.jpg	V	V
5	5.jpg	V	V
6	6.jpg	V	V
7	7.jpg	V	V
8	8.jpg	V	V
9	9.jpg	V	V
10	10.jpg	V	V
11	11.jpg	V	V
12	12.jpg	V	V
13	13.jpg	V	V
14	14.jpg	V	V
15	15.jpg	V	V
16	16.jpg	V	V
17	17.jpg	V	-
18	18.jpg	V	V
19	19.jpg	V	V
20	20.jpg	V	V
21	21.jpg	V	V
22	22.jpg	V	V
23	23.jpg	V	V
24	24.jpg	V	V
25	25.jpg	V	V

26	26.jpg	V	V
27	27.jpg	V	V
28	28.jpg	V	V
29	29.jpg	V	V
30	30.jpg	V	V
31	31.jpg	V	V
32	32.jpg	V	V
33	33.jpg	V	V
34	34.jpg	V	V
35	35.jpg	V	V
36	36.jpg	V	V
37	37.jpg	V	V
38	38.jpg	V	V
39	39.jpg	V	V
40	40.jpg	V	V
41	41.jpg	V	V
42	42.jpg	V	V
43	43.jpg	V	V
44	44.jpg	V	V
45	45.jpg	V	V
46	46.jpg	V	V
47	47.jpg	V	V
48	48.jpg	V	V
49	49.jpg	V	V
50	50.jpg	V	V
51	51.jpg	V	V
52	52.jpg	V	V
53	53.jpg	V	V
54	54.jpg	V	V

55	55.jpg	V	V
56	56.jpg	V	V
57	57.jpg	V	V
58	58.jpg	V	V
59	59.jpg	V	V
60	60.jpg	V	V
61	61.jpg	V	V
62	62.jpg	V	V
63	63.jpg	V	V
64	64.jpg	V	V
65	65.jpg	V	V
66	66.jpg	V	V
67	67.jpg	V	V
68	68.jpg	V	V
69	69.jpg	V	V
70	70.jpg	V	V
71	71.jpg	V	V
72	72.jpg	V	V
73	73.jpg	V	V
74	74.jpg	V	V
75	75.jpg	V	V
76	76.jpg	V	V
77	77.jpg	V	V
78	78.jpg	V	V
79	79.jpg	V	V
80	80.jpg	V	V
81	81.jpg	V	V
82	82.jpg	V	V
83	83.jpg	V	V

84	84.jpg	V	V
85	85.jpg	V	V
86	86.jpg	V	V
87	87.jpg	V	-
88	88.jpg	V	V
89	89.jpg	V	V
90	90.jpg	V	-
91	91.jpg	V	-
92	92.jpg	V	V
93	93.jpg	V	V
94	94.jpg	V	V
95	95.jpg	V	V
96	96.jpg	V	V
97	97.jpg	V	V
98	98.jpg	V	V
99	99.jpg	V	V
100	100.jpg	V	V
101	101.jpg	V	V
102	102.jpg	V	V
103	103.jpg	V	V
104	104.jpg	V	V
105	105.jpg	V	V
106	106.jpg	V	V
107	107.jpg	V	V
108	108.jpg	V	V
109	109.jpg	V	V
110	110.jpg	V	V
111	111.jpg	V	V
112	112.jpg	V	V

113	113.jpg	V	V
114	114.jpg	V	V
115	115.jpg	V	V
116	116.jpg	V	V
117	117.jpg	V	V
118	118.jpg	V	V
119	119.jpg	V	V
120	120.jpg	V	V
121	121.jpg	V	V
Jumlah		121	117

Berdasarkan tabel 5.3 dapat dilihat bahwa banyaknya data dapat mempengaruhi hasil pengujian atau akurasi dari proses deteksi dan pengenalan. Dari jumlah total 121 data citra yang memuat objek rambu lalu lintas didapat 121 citra berhasil terdeteksi menggunakan metode segmetasi warna *hue*, *saturation*, *value* dan 117 citra berhasil dikenali menggunakan metode *template matching*. Berikut adalah perhitungan akurasi dan *error rate* dalam proses deteksi objek rambu lalu lintas sebesar :

$$Akurasi = \frac{121}{121} \times 100\% = 100\%$$

$$Error\ rate = \frac{0}{121} \times 100\% = 0\%$$

Berikut ini adalah perhitungan akurasi dan *error rate* pada proses pengenalan objek rambu lalu lintas sebesar :

$$Akurasi = \frac{117}{121} \times 100\% = 96.69\%$$

$$Error\ rate = \frac{4}{121} \times 100\% = 3.305\%$$

Tabel 5.4 berikut adalah pengelompokan data menurut jenis dan informasinya. Dari 121 data citra dikelompokkan menjadi 28 jenis data berdasarkan informasi. Untuk lebih detail keseluruhan data bisa dilihat pada tabel dibawah :

Tabel 5.4 Pengelompokan data citra rambu lalu lintas berdasarkan informasi

No	Jenis Rambu	Informasi	Jumlah Data
1	Peringatan	Area Pejalan Kaki dan Anak Kecil	9
2	Peringatan	Belokan Tajam ke Kanan	4
3	Peringatan	Bundaran	4
4	Peringatan	Hati-Hati	6
5	Peringatan	Jalan Belokan Ganda	3
6	Peringatan	Jalan Cembung	5
7	Peringatan	Jalan Menyatu	3
8	Peringatan	Jalan Menyerong Dua Arah	3
9	Peringatan	Simpang Tiga Kanan Kiri	3
10	Peringatan	Simpang Tiga Ganda	3
11	Peringatan	Jembatan	5
12	Peringatan	Pejalan Kaki	6
13	Peringatan	Penggabungan Jalan	3
14	Peringatan	Penyempitan Jalan	2
15	Peringatan	Penyempitan Jalan Sebelah Kiri	3
16	Peringatan	Rel Kereta Api dengan Palang Pintu	3
17	Peringatan	Rel Kereta Api tanpa Palang Pintu	4
18	Peringatan	Simpang Empat	7
19	Peringatan	Simpang Tiga Kanan	6
20	Peringatan	Simpang Tiga Kiri	13

21	Peringatan	Simpang Tiga Lengan Serong	2
22	Peringatan	Tikungan Tajam ke Kiri	3
23	Peringatan	Belokan ke Kiri	4
24	Peringatan	Traffic Light	3
25	Perintah	Wajib Lajur Kiri	3
26	Perintah	Wajib Mengitari Bundaran	3
27	Perintah	Dimohon Berhenti	4
28	Larangan	Dilarang Masuk	4
Jumlah Seluruh Data			121

Tabel 5.5 Pengaruh Jumlah Data Acuan Terhadap Tahap Pengujian Pengenalan dan Prosentase Akurasinya

Data Citra Acuan	Data Citra Uji	Data Terkenali	Data Tidak Terkenali	Akurasi
121 Citra (100%)	121 Citra (100%)	121 Citra	0 Citra	100%
110 Citra (90%)	11 Citra (20%)	117 Citra	4 Citra	96.69%
85 Citra (70%)	36 Citra (30%)	103 Citra	18 Citra	85.12%

Berdasarkan Tabel 5.5 dapat dilihat bahwa banyaknya data citra acuan dapat mempengaruhi tingkat akurasi pengujian. Data pertama yaitu data acuan 121 citra rambu lalu lintas dan data uji 121 citra rambu lalu lintas menghasilkan akurasi paling tinggi yaitu 100%. Untuk data kedua yaitu data acuan 110 citra rambu lalu lintas dan data uji 11 citra rambu lalu lintas tingkat akurasinya menurun yaitu 96.69%. Untuk data ketiga tingkat akurasinya hanya sebesar 85%. Dari tabel diatas bisa disimpulkan untuk sistem pengenalan terbaik yaitu dengan menggunakan 100% data citra acuan dan 100% data citra uji yang menghasilkan tingkat akurasi tertinggi.

5.9 Pembahasan Perhitungan Jarak

Pembahasan perhitungan jarak pada sistem pengenalan menggunakan perhitungan jarak *euclidean* dan jarak *normalized euclidean*. Perhitungan ini dilakukan dengan rumus *euclidean* dan dicari jarak yang paling minimum antara citra template acuan dengan citra uji. Rumus *euclidean* sebagai berikut [13] :

$$d(\bar{x}, \bar{y}) = \sqrt{\sum_{i=1}^n (\bar{x}_i - \bar{y}_i)^2}$$

\bar{x} = Fitur vector ciri citra *template* acuan

\bar{y} = Fitur vektor ciri citra uji

$d(\bar{x}, \bar{y})$ = jarak *Euclidean* antara x dan y

\bar{x}_{ij} = Nilai fitur vektor ke-i dari citra x

\bar{y}_{ij} = Nilai fitur vektor ke-i dari citra y

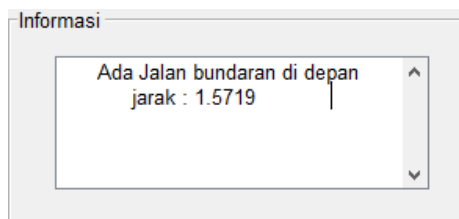
n = Banyaknya komponen i



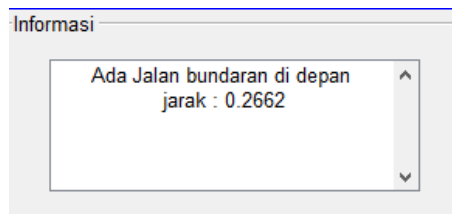
Gambar 5.13 Citra input



Gambar 5.14 Citra kandidat deteksi (kiri) dan citra template acuan (kanan)



Gambar 5.14 Hasil penampilan informasi dan perhitungan jarak *euclidean* antara citra template acuan dan citra kandidat



Gambar 5.15 Hasil penampilan informasi dan perhitungan jarak *normalized euclidean* antara citra template acuan dan citra uji

Dalam hal ini citra kandidat terdeteksi dan citra template acuan telah di ekstraksi fitur vektornya dan telah di hitung jarak *euclidean*. Penjelasannya sebagai berikut,

Fitur vector citra template acuan :

83.5786235	86.8109	0.28888	69.8896	71.6258	0.2673	38.6995	44.2497	0.96422
------------	---------	---------	---------	---------	--------	---------	---------	---------

Fitur vector citra uji terdeteksi :

83.9009	87.1582	0.2904	70.4004	72.1231	0.2846	39.5196	45.2818	1.0916
---------	---------	--------	---------	---------	--------	---------	---------	--------

Jarak *euclidean* :

$$\begin{aligned}
 d_e(\bar{x}, \bar{y}) &= \sqrt{\sum_{i=1}^n (\bar{x}_i - \bar{y}_i)^2} \\
 &= \sqrt{(83.57 - 83.90)^2 + (86.81 - 87.15)^2 + \dots + (0.96 - 1.09)^2} \\
 &= 1.577
 \end{aligned}$$

Jarak *normalized euclidean* :

$$\begin{aligned}
 E &= \frac{83.57 + 86.81 + \dots + 0.96}{9} = 44.04 \\
 \sigma &= \sqrt{\frac{(83.57 - 44.04)^2 + (86.81 - 44.04)^2 + \dots + (0.96 - 44.04)^2}{9}} \\
 &= 34.2152 \\
 d_{ne}(\bar{x}, \bar{y}) &= \sqrt{\sum_{i=1}^n \frac{(\bar{x}_i - \bar{y}_i)^2}{\sigma_k}} \\
 &= \sqrt{\frac{(83.57 - 83.90)^2}{34.2152} + \frac{(86.81 - 87.15)^2}{34.2152} + \dots + \frac{(0.96 - 1.09)^2}{34.2152}} \\
 &= 0.2696
 \end{aligned}$$

5.10 Pembahasan Hasil Pengujian

Pembahasan hasil pengujian difokuskan pada hasil pengujian dan proses pengenalan citra objek rambu lalu lintas. Pada tugas akhir ini, pembahasan hasil pengujian digunakan untuk mengetahui kinerja sistem pengenalan citra rambu lalu lintas menggunakan metode *template matching*. Dari hasil yang telah didapatkan pada subbab sebelumnya, sistem mampu mengenali citra objek rambu lalu lintas dengan cukup baik. Pada subbab ini akan dijelaskan tentang cara menghitung tingkat akurasi sistem dengan menggunakan persamaan berikut,

$$Akurasi = \frac{Jumlah\ citra\ terkenali}{jumlah\ seluruh\ citra} \times 100$$

5.11 Pembahasan Penyebab Besar Kecilnya Akurasi

Penyebab utama rendahnya akurasi sistem pengenalan citra rambu lalu lintas dengan metode *template matching* adalah :

1. Terletak pada kualitas citra yang diakuisisi kurang begitu baik. Hal tersebut dikarenakan jarak jauh atau dekatnya pengambilan citra melalui kamera terhadap objek rambu dan perbedaan intensitas cahaya yang tidak merata. Sehingga proses segmentasi kurang maksimal.
2. Terletak pada sudut pengambilan citra yang kurang pas terhadap objek rambu yang menyebabkan citra sulit untuk terdeteksi.
3. Jumlah banyaknya data citra template acuan, semakin banyak data citra template acuan yang diekstraksi dan dijadikan dataset, maka semakin banyak informasi yang tersimpan. Sehingga ketika dilakukan pengujian, akurasi yang diberikan lebih tinggi.

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisi tentang beberapa kesimpulan yang dihasilkan berdasarkan penelitian yang telah dilaksanakan. Di samping itu, pada bab ini juga dimasukkan beberapa saran yang dapat digunakan jika penelitian ini ingin dikembangkan.

6.1 Kesimpulan

Berdasarkan analisis terhadap hasil pengujian yang telah dilakukan terhadap sistem pengenalan citra digital rambu lalu lintas dengan metode *template matching*, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Tugas akhir ini telah berhasil melakukan deteksi citra objek rambu lalu lintas menggunakan metode segmentasi warna *hue, saturation, value* dan pengenalan citra objek rambu lalu lintas menggunakan metode *template matching*.
2. Metode deteksi segmentasi warna *hue, saturation, value* pada tugas akhir ini dapat melakukan deteksi dengan tingkat akurasi 100% sedangkan metode pengenalan dengan metode *template matching* dapat melakukan pengenalan dengan tingkat akurasi terbaik sebesar 100%. Pengujian pengenalan dengan hasil terbaik ini dilakukan pada data acuan 121 citra rambu lalu lintas dan data uji 121 citra rambu lalu lintas.

6.2 Saran

Berdasarkan hasil yang dicapai pada penelitian ini, ada beberapa hal yang penulis sarankan untuk pengembangan selanjutnya yaitu:

1. Sebaiknya dilakukan penambahan jumlah data citra objek rambu lalu lintas khususnya selain rambu yang menjadi citra *template* acuan pada tugas akhir ini yang berguna untuk memaksimalkan nilai akurasi sistem.

2. Beberapa hal yang ada kaitannya dengan teknik fotografi seperti halnya, pencahayaan, jarak pengambilan gambar, sudut(*angle*) pengambilan gambar terhadap objek, dan sebagainya perlu dipertimbangkan untuk mendapatkan data citra yang lebih baik kualitasnya karena citra masukkan sangatlah berpengaruh saat diproses dalam sistem.
3. Penerapan metode pengenalan lainnya perlu dipertimbangkan sebagai pembandingan agar diperoleh hasil yang lebih baik seperti contohnya penerapan metode jaringan syaraf tiruan (*neural network*) untuk metode pengenalan citra rambu lalu lintas.

DAFTAR PUSTAKA

- [1] Al-Habib, Hasanuddin, Budi Setiyono. 2013. *Super Resolusi Objek Berbasis Citra Tracking Menggunakan Metode Phased Based Image Matching dan Metode Proyeksi Pada Himpunan Convex*. Jurnal Sains dan Seni POMITS, 2(1): 1-6.
- [2] Aryani, Mira, Nurul Hidayat. 2007. *Pengenalan Gambar Lalu Lintas Menggunakan Metode Template Matching*. Surabaya. Tugas Akhir - Institut Teknologi Sepuluh November.
- [3] Broggi, Alberto, Pietro Cerri, Paolo Medici, dan Pier Paolo Porta. 2007. *Real Time Road Sign Recognition*. Proceeding of the 2007 IEEE Intelligent Vehicles Symposium Istanbul, Turkey, June 13-15, 2007.
- [4] Brunelli, Roberto. 2009. *Template Matching Techniques in Computer Vision Theory and Practice*. Italia: Wiley.
- [5] Gonzalez, Rafael, C, Richard E. Woods, dan Steven L. Eddins. 2008. *Digital Image Processing*. New Jersey: Pearson Prentice Hall.
- [6] Leksono, Bowo, Achmad Hidayanto dan R. Rizal Isnanto. 2011. *Aplikasi Metode Template Matching untuk Klasifikasi Sidik Jari*. <http://ejournal.undip.ac.id/index>, TRANSMISI, 13(1): 1-6.
- [7] Li, Haojie, Fuming Sun, Lijuan Liu, dan Ling Wang. 2015. *A novel traffic sign detection method via color segmentation and robust shape matching*. Dalian University of Technology, Dalian, Liaoning, China.

- [8] Maldonado-Bascon, Saturnino, Hilario Gomes-Moreno. 2007. *Road-Sign Detection and Recognition Based on Support Vector Machines*. IEEE Transaction on Intelligent Transportation Systems, Vol. 8(2): 264-277.
- [9] Mathias, Markus, Radu Timofte, Rodrigo Benenson, dan Luc Van Gool. 2013. *Traffic Sign Detection – How far are we from the solution?*. Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN 2013) Dallas, USA, August 2013.
- [10] McAndrew, Alasdair. 2004. *An Introduction to Digital Image Processing with MATLAB*. Alasdair McAndrew School of Computer Science and Mathematics Victoria University of Technology.
- [11] Prasetyo, Eko. 2011. *Pengolahan Citra Digital dan Aplikasinya menggunakan MATLAB*. Gresik. Andi Offset Yogyakarta.
- [12] Pratama, Andhika, Izzati Muhimmah. 2011. *Aplikasi Pengenalan Rambu Berbentuk Belah Ketupat*. Seminar Nasional Aplikasi Teknologi Informasi 2011, Yogyakarta, 17-18 Juni 2011.
- [13] Rafael, Gonzales, Richard E. Woods. 2002. *Digital Image Processing*. United States of America: Tom Robbins Publisher.
- [14] Rinanto, Lutfi, Aris Sugiharto dan Indriyati. 2013. *Aplikasi Pendeteksi Objek Lingkaran Pada Citra Dengan Transformasi Hough*. Journal of Informatics and Technology, 2(4): 1-9.

- [15] Sallah, Siti Sarah Md, Fawnizu Azmadi Hussin, dan Mohd Zuki Yusoff. 2010. *Shape-based Road Sign Detection and Recognition for Embedded Application Using MATLAB*. Proceeding of International Conference on Intelligent and Advanced Systems(CIAS 2010), Bandar Seri Iskandar, Malaysia.
- [16] Shelke, Chetan J, Pravin Karde. 2012. *Traffic Sign Recognition*. International Journal of Computanional Engineering Research, 2(8): 47-52.
- [17] Soendoro, David, Iping Supriana. 2011. *Traffic Sign Recognition with Color-based Method, Shape-arc Estimation and SVM*. Proceedings of International Conference on Electrical Engineering and Informatics, 17-19 July 2011, Bandung, Indonesia.
- [18] Souki, Mohammad Amin, Lotfi Boussaid dan M.Abid. 2008. *An Embedded System for Real-Time Traffic Sign Recognizing*. 3rd International Design and Test Workshop, IDT 2008, Monastir, 20-22 Desember 2008, pp. 273-276.
- [19] Waite, Sheldon, Erdal Oruklu. 2013. *FPGA-Based Traffic Sign Recognition for Advance Driver Assistance System*. Journal of Transportation Technologies, 3(1): 1-16.

BIODATA PENULIS



Penulis memiliki nama lengkap Yusuf, lahir di Mojokerto pada tanggal 12 Juli 1995. Penulis berasal dari Kota Jombang, di Perum. Sambong Indah blok D/23, Jl. Laksda Adi Sucipto, Kab. Jombang. Pendidikan formal yang pernah ditempuh yaitu MIN Kauman Utara Jombang, SMP Negeri 1 Jombang, dan SMAN 3 Jombang. Kemudian, penulis melanjutkan studi di jurusan Matematika ITS, dengan bidang minat ilmu komputer. Dalam bidang minat ini penulis mulai mengenal bahasa pemrograman diantaranya adalah C, C++, Java, PHP-MySQL, dan MATLAB. Semasa menempuh jenjang pendidikan S-1, penulis juga aktif dalam kegiatan non-akademis diantaranya aktif di organisasi kemahasiswaan Matematika ITS dan mengikuti kepanitiaan acara besar yang ada di ITS diantaranya: OMITS, dan Panitia INTERN FMIPA. Selama penulisan Tugas Akhir ini Penulis tidak lepas dari kekurangan, untuk itu penulis mengharapkan kritik, saran, dan pertanyaan mengenai Tugas Akhir ini yang dapat dikirimkan melalui *e-mail* : mr.yusuf.mustafa@gmail.com.